

AD-A064 417

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/6 20/13  
INVESTIGATION OF ACCELERATING THE FINITE ELEMENT SOLUTION OF TH--ETC(U)  
DEC 78 F J JAEGER

UNCLASSIFIED

AFIT/ONE/PH/78D-6

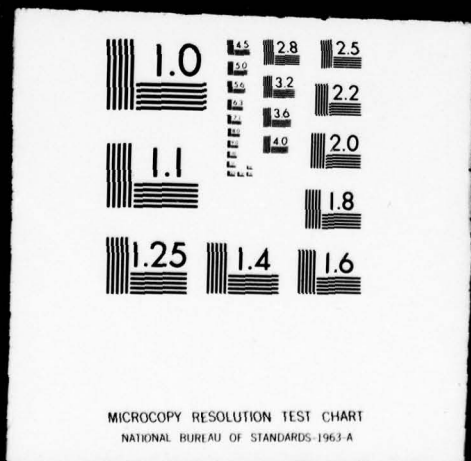
NL

OF 2  
AD A064 417



1 OF 2

AD  
A064417



ADA064417

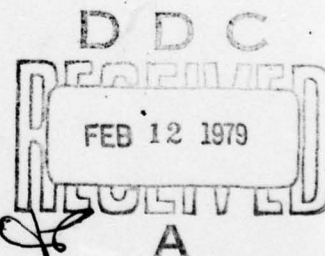
DDC FILE COPY

INVESTIGATION OF ACCELERATING THE  
FINITE ELEMENT SOLUTION OF THE  
TWO DIMENSIONAL STEADY STATE  
HEAT TRANSFER EQUATION

THESIS

AFIT/GNE/PH/78D-6

Frederick J. Jaeger  
Captain USAF



Approved for public release; distribution unlimited

79 01 30 119

14

AFIT/GNE/PH/78D-6

6

INVESTIGATION OF ACCELERATING THE FINITE  
ELEMENT SOLUTION OF THE TWO DIMENSIONAL  
STEADY STATE HEAT TRANSFER EQUATION •

9

Master's Thesis

THESIS

11

Dec 78

Presented to the Faculty of the School of Engineering of  
the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

12 97 P'

10

by

Frederick J. Jaeger ~~B.S.M.E.~~

Captain

USAF

Graduate Nuclear Engineering

Approved for public release; distribution unlimited.

012 225

elt

Preface

This study was originally limited to different scans and the resultant reordering of equations. This technique for changing convergence rates had not been explored with the equations resulting from finite element solutions. Scannings had been shown to accelerate finite difference solutions and I planned to extend these results to a finite element solution. The discovery of the coarse mesh rebalancing method changed the direction of the study. The coarse mesh rebalancing promised an opportunity to accelerate the solutions without finding the optimum overrelaxation factors. The number of scans to be tested was reduced and coarse mesh rebalancing as function of overrelaxation factor, rebalancing frequency, and coarse mesh size became the bulk of the study.

I wish to thank Dr. Bernard Kaplan of the Physics Department of the Air Force Institute of Technology for his constant guidance during this study. I also wish to thank Dr. W. Kessler of the Air Force Materials Laboratory for sponsoring this study.

Frederick J. Jaeger

AFIT		DATE SUBMITTED
DOC		DATE SIGNED
WHEN FORWARDED		<input type="checkbox"/>
INVESTIGATION		<input type="checkbox"/>
BY		
DISTRIBUTION/AVAILABILITY CODES		
UNCL	AVAIL. DOC/OT SPECIAL	
A		

## Contents

	Page
Preface . . . . .	ii
List of Figures . . . . .	v
List of Tables . . . . .	vi
Abstract . . . . .	vii
I. Introduction . . . . .	1
Background . . . . .	1
Purpose . . . . .	2
Scope . . . . .	2
Plan of Development . . . . .	3
II. Theory . . . . .	4
The Partial Differential Equation . . . . .	4
The Finite Element Method . . . . .	4
Boundary Conditions . . . . .	16
Iteration Method . . . . .	17
Overrelaxation . . . . .	18
Scanning . . . . .	19
Coarse Mesh Rebalancing . . . . .	20
Partitioning . . . . .	25
Weighting Vectors . . . . .	25
III. Procedure . . . . .	26
Computer . . . . .	26
Problem . . . . .	26
Programming . . . . .	28
Convergence Criteria . . . . .	29
Trial Vector . . . . .	29
Scanning . . . . .	30
Serial Scan . . . . .	30
Modified Serial Scan . . . . .	30
Spiral Scan . . . . .	33
Coarse Mesh Rebalancing . . . . .	33
IV. Results . . . . .	37
Scanning . . . . .	37
Coarse Mesh Results . . . . .	42

	Page
V. Conclusions and Recommendations . . . . .	60
Conclusions . . . . .	60
Recommendations . . . . .	61
Bibliography . . . . .	62
Appendix A: Sample Coarse Rebalancing . . . . .	63
Appendix B: Plots of Maximum Error vs Number of Iterations .	66

### List of Figures

Figure		Page
1	Domain Divided into Finite Elements . . . . .	5
2	Typical Finite Element . . . . .	5
3	Coarse and Fine Meshes . . . . .	21
4	Finite Element Grid . . . . .	27
5	Isosceles Right Elements . . . . .	27
6	Serial Scanning . . . . .	29
7	Matrix for Serial Scan . . . . .	29
8	Modified Serial Scanning . . . . .	30
9	Matrix for Mod-Serial Scan . . . . .	30
10	Spiral Scanning . . . . .	32
11	Matrix for Spiral Scanning . . . . .	32
12	Graphic Determination of Optimum Overrelaxation Factor for the Serial Scan . . . . .	38
13	Graphic Determination of Optimum Overrelaxation Factor for the Modified Serial Scan . . . . .	39
14	Graphic Determination of Optimum Overrelaxation Factor for the Spiral Scan . . . . .	40
15	Effect of Rebalancing . . . . .	51
16	Effect of Rebalancing . . . . .	52
17	Effect of Rebalancing . . . . .	53
18	Effect of Rebalancing . . . . .	54
19	Effect of Rebalancing . . . . .	55
20	Effect of Rebalancing . . . . .	56
21	Effect of Rebalancing . . . . .	57
22	Effect of Rebalancing . . . . .	58

List of Tables

Table		Page
I	Scanning Results . . . . .	41
II	Coarse Mesh Results: Overrelaxation Factor = 1.50 .	43
III	Coarse Mesh Results: Overrelaxation Factor = 1.60 .	44
IV	Coarse Mesh Results: Overrelaxation Factor = 1.70 .	45
V	Coarse Mesh Results: Overrelaxation Factor = 1.80 .	46
VI	Coarse Mesh Results: Overrelaxation Factor = 1.84 .	48

Abstract

The relative solution time is studied for two methods of accelerating successive overrelaxation. Reordering of equations by nodal point scanning, and coarse mesh rebalancing are used. The finite element solution of the steady-state two-dimensional heat transfer equation is used to test these methods.

Scanning the boundary nodal points first was found to reduce the number of iterations necessary for convergence by up to 13%, but computer execution time was increased by up to 7%. Coarse mesh rebalancing was found to speed the solution with arbitrary successive overrelaxation factor by reducing the number of iterations to 15% of that without rebalancing. The computer execution time required for a well chosen coarse mesh was only reduced to 30% of that without rebalancing. The successive overrelaxation factor was found to influence the optimum rebalancing frequency. Solutions as fast as the solution with the optimum overrelaxation factor were obtained with rebalancing and arbitrary overrelaxation factors. Rules for a proper coarse mesh and rebalancing frequency are given.

INVESTIGATION OF ACCELERATING THE  
FINITE ELEMENT SOLUTION OF THE  
TWO DIMENSIONAL STEADY STATE  
HEAT TRANSFER EQUATION

I. Introduction

Background

Few mathematical problems have straightforward exact solutions in analytical form. A good approximate solution must be used because the exact solution is not available. The digital computer can find an approximate but relatively accurate solution using numerical methods.

The sponsor of this thesis, the Air Force Materials Laboratory, studies the thermal response and ablation characteristics of rocket nozzles and re-entry vehicles. These problems require solution of transient heat conduction equations. The transient heat conduction equation can be approximated by a steady-state heat conduction problem and numerical methods that assume steady-state conditions over certain time intervals.

This thesis is an attempt to reduce the time necessary to obtain one type of numerical solution to the steady-state heat conduction equation.

There are two common approximation techniques: finite elements, and finite differences. The acceleration of the finite difference method of solving this type of problem has already been studied by Pearson (Ref 1), Cudahy (Ref 2), and Wright (Ref 3). The finite ele-

ment method has been receiving more attention in recent years, but acceleration of this method has not received the study that the finite difference method had been given. This thesis will apply one of the accelerating techniques of Pearson, Cudahy, and Wright.

The finite element method, like the finite difference method, requires the solution of simultaneous equations. One equation is written for each nodal point in the region over which a solution is desired.

These simultaneous equations can be solved expediently by iteration. The equations are solved repeatedly with trial solutions, which are continually improved until two successive solutions are approximately equal. The iteration process is then said to have converged.

#### Purpose

This study is an attempt to reduce the number of iterations and the associated computer time to produce convergence. There are many well known methods of increasing convergence rates. Successive overrelaxation is, in general, the most successful and easiest to apply. Different scannings of the nodal points to reorder the set of equations can change the rate of convergence for successive overrelaxation. Another method that can change the rate of convergence for successive overrelaxation is the coarse mesh rebalancing method used by Nakamura (Ref 4).

These methods will be used and compared in this study.

#### Scope

Successive overrelaxation is used to solve the finite element approximation to the steady-state heat conduction equation in a

two-dimensional square. Three scanning schemes are used to compare changes in the rate of convergence of successive overrelaxation.

Coarse mesh rebalancing is also used to accelerate the successive overrelaxation iteration. Coarse mesh rebalancing is applied with four different coarse mesh sizes, five different successive overrelaxation factors, and six different rebalancing frequencies. The rates of convergence are compared for these solutions.

#### Plan of Development

The optimum overrelaxation factor is found for each scan experimentally. This requires solving the problem many times with different overrelaxation factors until the factor that required the fewest iterations is determined.

The number of iterations for each scan to converge using the optimum successive overrelaxation factors will be compared. The computer execution time will also be compared for these solutions.

The coarse mesh rebalancing method will be applied with different overrelaxation factors, different coarse mesh sizes, and different rebalancing frequencies for a single scan. These solutions will be compared with each other and with the results from the different scannings.

## II. Theory

### The Partial Differential Equation

The governing partial differential equation for steady-state, two-dimensional heat conduction for an isotropic homogeneous material is

$$\frac{k}{rc} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \frac{\dot{q}}{rc} = 0 \quad (1)$$

or

$$k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \dot{q} = 0 \quad (1A)$$

where  $T$  is the temperature in degrees Centigrade ( $^{\circ}\text{C}$ )

$k$  is the conductivity of the material in calories/cm-sec- $^{\circ}\text{C}$

$r$  is the density of the material in grams/centimeter

$c$  is the heat capacity of the material in calories/gm- $^{\circ}\text{C}$

$\dot{q}$  is the heat generation rate in calories/cm-sec

The solution of this equation defines the temperature, under steady-state conditions, at any point in the material.

### The Finite Element Method

The finite element method uses numerical approximation methods to solve an integral. The integral is divided over its domain into subdomains which are called finite elements (Fig 1).

Each element consists of nodal points connected by line segments. The elements are connected at nodal points and along the element boundaries (Fig 2).

The elements usually have straight boundaries and if the problem

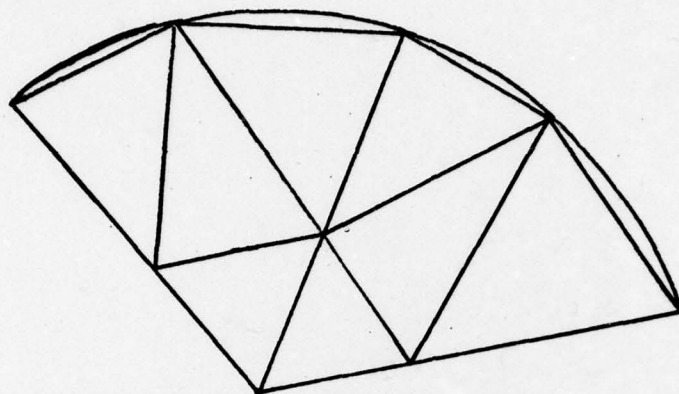


Figure 1. Domain Divided into Elements

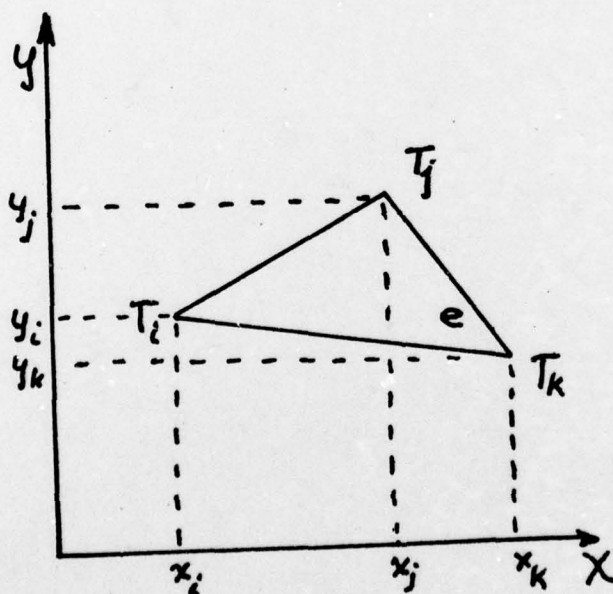


Figure 2. Typical Finite Element

domain has curved boundaries, then the curve is easily represented by a series of straight segments. This is an additional approximation that can be made with this method.

The differential equation for heat conduction is in differential form not in integral form. The integral form must be formulated before the finite element method can be applied (Ref 5).

The integral form of a governing differential equation can sometimes be related to the physical principles of the problem. In this case, it cannot be done readily, but variational calculus can provide the desired integral with or without a physical analog.

The problem can be cast as the minimization of an integral (Ref 5). That is, to find a function,  $u(z)$ , that minimizes the integral  $I$ .

$$I = \int_z F(z, u(z), u'(z)) dz \quad (2)$$

where  $z$  is the independent variable

$u$  is the function of  $z$

$u'$  is the derivative of  $u$  with respect to  $z$

$F$  is a functional of  $z$ ,  $u(z)$ , and  $u'(z)$

For  $I$  to be a minimum

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial z} \left( \frac{\partial F}{\partial u'} \right) = 0 \quad (3)$$

The boundary conditions for equation (3) are required to be the same as for the original problem (Ref 5:280).

The two-dimensional, steady-state heat conduction equation can be formulated in the same manner as equation (2)

$$I = \int_x \int_y F(x, y, T, T_x, T_y) dx dy \quad (4)$$

where  $T_x$  is the derivative of  $T$  with respect to  $x$

$T_y$  is the derivative of  $T$  with respect to  $y$

For the integral in equation (4) to be minimized, the following must be true.

$$\frac{\partial F}{\partial T} - \frac{\partial}{\partial x} \frac{\partial F}{\partial T_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial T_y} = 0 \quad (5)$$

Equation (1A) can be compared to equation (5) and a functional  $F$  determined by inspection. The function is composed of three parts.

$$\frac{\partial F}{\partial T} = q \quad (6)$$

$$- \frac{\partial}{\partial x} \frac{\partial F}{\partial T_x} = k \frac{\partial}{\partial x} \left( \frac{\partial T}{\partial x} \right) \quad (7)$$

$$- \frac{\partial}{\partial y} \frac{\partial F}{\partial T_y} = k \frac{\partial}{\partial y} \left( \frac{\partial T}{\partial y} \right) \quad (8)$$

Equations (7) and (8) are integrated with respect to  $x$  and  $y$  respectively to yield:

$$\frac{\partial F}{\partial T_x} = k \frac{\partial T}{\partial x} \quad (9)$$

$$\frac{\partial F}{\partial T_y} = -k \frac{\partial T}{\partial y} \quad (10)$$

Equation (6) is integrated with respect to  $T$ , equation (9) is integrated with respect to  $x$ , and equation (10) is integrated with respect to  $y$ . Each of these integrations yields an expression for  $F$ .

$$F = -\frac{1}{2} k \frac{\partial T}{\partial x} + \alpha(T) + \beta(T_y) \quad (11)$$

$$F = -\frac{1}{2} k \frac{\partial T}{\partial y} + \gamma(T) + \epsilon(T_x) \quad (12)$$

$$F = \dot{q} T + \zeta(T_x) + \eta(T_y) \quad (13)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\epsilon$ ,  $\zeta$ , and  $\eta$  are functions to be determined.

Because the three expressions for  $F$  must be equal, the unknown functions can be determined by comparison of equations (11), (12), and (13).

$$F = \dot{q} T - \frac{k}{2} \frac{\partial T}{\partial x} - \frac{k}{2} \frac{\partial T}{\partial y} \quad (14)$$

The functional  $F$  can now be substituted into equation (4) and (4) differentiated with respect to  $T$  and set equal to zero to obtain a minimum.

$$\frac{\partial I}{\partial T} = \frac{\partial}{\partial T} \left[ \iint_x \left[ -\frac{k}{2} \left( \frac{\partial T}{\partial x} \right) - \frac{k}{2} \left( \frac{\partial T}{\partial y} \right) + \dot{q} T \right] dx dy \right] \quad (15)$$

Equation (15) can be separated into two integrals for convenience.

$$\frac{\partial I}{\partial T} = \frac{\partial I_k}{\partial T} + \frac{\partial I_q}{\partial T} = 0 \quad (16)$$

where

$$\frac{\partial I_k}{\partial T} = \frac{\partial}{\partial T} \left[ \int_x \int_y \left[ -\frac{k}{2} \frac{\partial T}{\partial x} - \frac{k}{2} \frac{\partial T}{\partial y} \right] dx dy \right] \quad (17)$$

$$\frac{\partial I_q}{\partial T} = \frac{\partial}{\partial T} \left[ \int_x \int_y \dot{q} T \, dx dy \right] \quad (18)$$

These integrals will be evaluated numerically by dividing the domain into many subdomains and approximating the integrals over each subdomain or element and summing the individual results. This is the finite element method (Ref 6:85). The summations can also be made for the derivatives of the integrals.

$$\frac{\partial I_k}{\partial T} = \sum_i^E \frac{\partial I_k^{(e)}}{\partial T} \quad (19)$$

$$\frac{\partial I_q}{\partial T} = \sum_i^E \frac{\partial I_q^{(e)}}{\partial T} \quad (20)$$

where there are E finite elements.

For each element (Fig 2) with nodes, i, j, and k at  $(x_i, y_i)$ ,  $(x_j, y_j)$ , and  $(x_k, y_k)$ , respectively, integral over that element will be functions of the nodal temperatures,  $T_i$ ,  $T_j$ ,  $T_k$ , only (Ref 5, 6).

$$\frac{\partial I_k^{(e)}}{\partial T} = \underline{D}^{(e)} \frac{\partial I_k^{(e)}}{\partial \underline{t}^{(e)}} \quad (21)$$

$\underline{t}^{(e)}$  is a vector of the nodal temperatures for element  $e$ .

$$\underline{t}^{(e)} = \begin{bmatrix} T_i \\ T_j \\ T_k \end{bmatrix} \quad (22)$$

$\underline{D}^{(e)}$  is a matrix that locates the nodal points of element  $e$ .

$$\underline{D}^{(e)} = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} i \text{ row} \\ j \text{ row} \\ k \text{ row} \end{matrix} \quad (23)$$

Within each element it is assumed that temperature varies linearly in both  $x$  and  $y$  directions. This can be expressed as

$$\underline{t}^{(e)} = C_1^{(e)} + C_2^{(e)} x + C_3^{(e)} y \quad (24)$$

or if

$$\underline{p} = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \quad (25)$$

and

$$\underline{C}^{(e)} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad (26)$$

$$\underline{t}^{(e)} = \underline{p}^T \underline{C}^{(e)} \quad (27)$$

If  $\underline{t}^{(e)}$  is defined by equation (22), then equation (24) could be written for each component of  $\underline{t}^{(e)}$ . These equations could be

formed into a matrix equation like

$$\underline{t}^{(e)} = \underline{P}^{(e)} \underline{c}^{(e)} \quad (28)$$

$\underline{P}^{(e)}$  is defined as the position matrix for element  $e$ .

$$\underline{P} = \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \quad (29)$$

The constants in equation (26) may be found by multiplying the inverse of  $\underline{P}^{(e)}$  by  $\underline{t}^{(e)}$  and substituting into equation (27) to find the temperature within an element.

$$T = \underline{P}^{(e)-1} \underline{t}^{(e)} \quad (30)$$

The derivatives of equation (30) with respect to both  $x$  and  $y$  are needed to substitute into an elemental form of equation (17).

$$\frac{\partial}{\partial x} T^{(e)} = \underline{P}_x^T \underline{P}^{(e)-1} \underline{t}^{(e)} \quad (31)$$

$$\frac{\partial}{\partial y} T^{(e)} = \underline{P}_y^T \underline{P}^{(e)-1} \underline{t}^{(e)} \quad (32)$$

$$L_k^{(e)} = \frac{k^{(e)}}{2} \iint_{x,y} \left[ \underline{P}_x^T \underline{P}^{(e)-1} \underline{t}^{(e)} \right]^2 + \left[ \underline{P}_y^T \underline{P}^{(e)-1} \underline{t}^{(e)} \right]^2 dx dy \quad (33)$$

The derivative with respect to temperature is now taken and because temperature in an element can be defined solely in terms of

the three nodal temperatures, the derivative can be taken with respect to these temperatures.

$$\frac{d I_k^{(e)}}{d \underline{t}^{(e)}} = \frac{k}{2} \int_x \int_y \left[ 2 \left( \underline{p}_x^T \underline{P}^{(e)-1} \underline{t}^{(e)} \right) \left( \underline{p}_x^T \underline{P}^{(e)-1} \right)^T + 2 \left( \underline{p}_y^T \underline{P}^{(e)-1} \underline{t}^{(e)} \right) \left( \underline{p}_y^T \underline{P}^{(e)-1} \right)^T \right] dx dy \quad (34)$$

This assumes constant thermal conductivity over each element. This is valid for elements of homogeneous materials and elements small enough that any change in conductivity with temperature can be neglected.

Equation (34) can be rearranged by matrix algebra and by realizing that  $\underline{P}^{(e)-1}$  and  $\underline{t}^{(e)}$  are independent of  $x$  and  $y$ , the following is obtained.

$$\frac{\partial I_k^{(e)}}{\partial \underline{t}^{(e)}} = k^{(e)} \left( \underline{P}^{(e)} \right)^T \iint \left[ \underline{p}_x \underline{p}_x^T + \underline{p}_y \underline{p}_y^T \right] dx dy \left( \underline{P}^{(e)-1} \underline{t}^{(e)} \right) \quad (35)$$

The required differentiation is now carried out.

$$\underline{p}_x = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (36)$$

$$\underline{p}_x^T = [0 \ 1 \ 0] \quad (37)$$

$$\underline{p}_x \underline{p}_x^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (38)$$

$$\underline{p}_y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (39)$$

$$\underline{p}_y^T = [0 \ 0 \ 1] \quad (40)$$

$$\underline{p}_y \underline{p}_y^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (41)$$

Equation (35) requires that equations (38) and (41) be added together.

$$\underline{p}_x \underline{p}_x^T + \underline{p}_y \underline{p}_y^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (42)$$

Substituting into equation (35), an integral that can be simply evaluated is obtained. The result is a function of the area of the element  $A^{(e)}$ .

$$\iint \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} dx \, dy = A^{(e)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (43)$$

The final resubstitution is made into equation (35).

$$\frac{\partial \underline{I}_k^{(e)}}{\partial \underline{t}^{(e)}} = \underline{K}^{(e)} (\underline{P}^{(e)})^{-1} \underline{A}^{(e)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underline{P}^{(e)-1} \underline{t}^{(e)} \quad (44)$$

The elemental conduction matrix  $\underline{K}^{(e)}$  can be defined from equation (44) such that:

$$\frac{\partial \underline{I}_k^{(e)}}{\partial \underline{t}^{(e)}} = \underline{K}^{(e)} \underline{t}^{(e)} \quad (45)$$

The total problem is formed by summing the elemental conduction matrices for all elements into a global conduction matrix.

$$\underline{K} = \sum_{e=1}^E \underline{D}^{(e)} \underline{K}^{(e)} \underline{D}^{(e)T} \quad (46)$$

The elemental heat generation vector is formed in a manner similar to the formation of the elemental conduction matrix. Recalling equation (18) and rewriting for a single element

$$\underline{I}_q^{(e)} = \int_x \int_y \underline{\dot{q}}^{(e)} \underline{t}^{(e)} dx dy \quad (47)$$

It is now assumed that heat generation is constant over each element. This assumption is also valid for elements of such size that temperature and distance effects can be neglected.

Equation (27) and (28) are rearranged and substituted into equation (47). The assumption that was just made allows the heat gener-

ation to be brought outside the integral.

$$\dot{I}_q^{(e)} = \dot{q}^{(e)} \int_x \int_y \underline{p}^T \underline{P}^{(e)-1} \underline{t}^{(e)} dx dy \quad (48)$$

The derivative with respect to  $\underline{t}^{(e)}$  is taken as in equation (34).

$$\frac{d \dot{I}_q^{(e)}}{d \underline{t}^{(e)}} = \dot{q}^{(e)} \int_x \int_y \underline{p}^T \underline{P}^{(e)-1} dx dy \quad (49)$$

Recalling that  $\underline{p}^{(e)}$  is independent of the variables  $x$  and  $y$ , it can be factored out of the integral when the terms are rearranged.

$$\frac{d \dot{I}_q^{(e)}}{d \underline{t}^{(e)}} = \dot{q}^{(e)} (\underline{P}^{(e)-1})^T \int_x \int_y \underline{p} dx dy \quad (50)$$

$$= \dot{q}^{(e)} (\underline{P}^{(e)-1})^T \int_x \int_y \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} dx dy \quad (51)$$

The integral can be carried out as the sum of three integrals, which can be recognized as the area, the centroid about the x-axis, and the centroid about the y-axis. After carrying out the integrations the following results

$$\frac{d \dot{I}_q^{(e)}}{d \underline{t}^{(e)}} = \dot{q}^{(e)} (\underline{P}^{(e)-1})^T \begin{bmatrix} 1 \\ \frac{x_i + x_j + x_k}{3} \\ \frac{y_i + y_j + y_k}{3} \end{bmatrix} A^{(e)} \quad (52)$$

After the multiplication has been carried out, the elemental heat generation vector can be defined.

$$\frac{d\bar{I}_q^{(e)}}{d\bar{t}^{(e)}} = \frac{\dot{q}^{(e)} A^{(e)}}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \underline{\bar{q}}^{(e)} \quad (53)$$

The global heat generation vector can be formed in the same way that the global heat conduction matrix was formed.

$$\underline{\bar{q}} = \sum_{e=1}^E \underline{D}^{(e)} \underline{\bar{q}}^{(e)} \quad (54)$$

The total problem has now been formed and can be stated as

$$\frac{d\bar{I}}{d\bar{t}} = \underline{K} \underline{\bar{t}} - \underline{\bar{q}} \quad (55)$$

or

$$\underline{K} \underline{\bar{t}} = \underline{\bar{q}} \quad (56)$$

#### Boundary Conditions

Generalized boundary conditions that specify a known temperature on part of the boundary and a known heat flux on the remainder of the boundary are easily handled. The convection boundary condition can also be handled but is not treated in this study.

After the finite mesh has been formulated, and the system of equations written, those nodal points that have known temperatures are eliminated from the temperature vector (Ref 5). The elimination

requires the addition of the known temperatures times the corresponding elements in the conduction matrix to the known heat generation vector. For example, if  $T_1$  were known

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad (57)$$

becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 4 & 1 \\ 0 & 4 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} T_1 \\ Q_2 - 2T_1 \\ Q_3 - 3T_1 \\ Q_4 - 4T_1 \end{bmatrix} \quad (58)$$

The first row and column of the matrix in equation (58) can now be eliminated.

The heat flux boundary condition is included in the problem by creating a vector of the specified heat fluxes and adding it to the heat generation vector. For example, if  $q_i$  were specified

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} + \begin{bmatrix} q_4 \\ c \\ 0 \end{bmatrix} = \underline{q} \quad (59)$$

This sum is then used as  $\underline{q}$  in equation (56).

#### Iteration Method

The method of successive displacements (Gauss-Seidel) replaces each component of the trial vector with a new value as soon as it can

be calculated. If the matrix is

$$\underline{A} \underline{x} = \underline{y} \quad (60)$$

where  $\underline{A}$  is a  $n \times n$  matrix

$\underline{x}$  is the unknown vector

$\underline{y}$  is the known vector

The matrix can be divided into three matrices such that

$$\underline{A} = \underline{L} + \underline{D} + \underline{U} \quad (61)$$

where  $\underline{L}$  has the same lower triangle as  $\underline{A}$  but is otherwise zero

$\underline{U}$  has the same upper triangle as  $\underline{A}$  but is otherwise zero

$\underline{D}$  has the elements of the main diagonal of  $\underline{A}$

The method of successive displacements can be written

$$\underline{x}^{n+1} = \underline{x}^n + \left[ \underline{y} - \underline{L}\underline{x}^{n+1} - \underline{D}\underline{x}^n - \underline{U}\underline{x}^n \right] \quad (62)$$

where the superscripts here indicate iteration number.

### Overrelaxation

If a multiplying factor is added to the successive displacement equation, the rate of convergence can be changed.

$$\underline{x}^{n+1} = \underline{x}^n + w \left[ \underline{y} - \underline{L}\underline{x}^{n+1} - \underline{D}\underline{x}^n - \underline{U}\underline{x}^n \right] \quad (63)$$

The process is called underrelaxation or overrelaxation, depending on whether the multiplying constant is less than one or greater than one, respectively. The multiplying factor can be chosen to increase the rate of convergence. Such a factor is, generally, greater than one, and equation (63) is commonly called successive overrelaxation, and

the multiplying factor is called the overrelaxation factor.

There exists an overrelaxation factor that will give the highest average rate of convergence. This factor is called the optimum overrelaxation factor. For the special case when the system of equations is written for a constantly ordered scan, on a unit square, and with uniform mesh spacings, the optimum overrelaxation factor can be predicted (Ref 7).

$$W_{OPT} = \frac{2}{1 + \sin(h\pi)} \quad (64)$$

where  $h$  is the spacing between nodes.

This equation can often be used to estimate the optimum overrelaxation factor.

#### Scanning

The average rate of convergence can be different for different ordering of equation (Ref 8:283). This effect has been explored for the equations resulting from the finite difference solutions to the heat conduction equation by Pearson (Ref 1), and Cudahy (Ref 2).

Both Pearson and Cudahy ordered their equations so that the temperatures at points near the known boundaries were calculated early in each iteration. The problem was then worked toward the center. This technique uses known information sooner than other possible equation orderings. The scans may or may not be consistent and, therefore, the optimum overrelaxation factor cannot always be predicted.

Some of the same orderings that Pearson and Cudahy used for finite differences will be used for finite elements.

### Coarse Mesh Rebalancing

Coarse mesh rebalancing can increase the rate of convergence by decaying the low frequency eigenvector components more efficiently than the iteration scheme alone (Ref 4). Low frequency eigenvectors are decayed most effectively, when the high frequency eigenvectors are small in comparison to the low frequency eigenvectors.

Each value of the unknown vector is normally calculated from the same equation in the iteration scheme. This requires that the components surrounding a nodal value be changed before that nodal value can be changed. One way of looking at coarse mesh rebalancing is the application of a more general view of the problem to each node. This can change a nodal value based on influential values that are some distance away without working through the nodes surrounding the node under consideration.

In coarse mesh rebalancing, the fine mesh is divided into  $L$  subregions (Fig 3). This is done and the trial vector is rebalanced by multiplying the components of the trial vector,  $\underline{t}_0$ , in each subregion by a corresponding rebalancing coefficient. These coefficients,  $\Phi_i$ , are determined by the coarse mesh procedure and their values will approach one as the problem converges. The weighted residual method is used to determine these coefficients (Ref 4). The rebalancing is carried out according to

$$\underline{t} = \left( \sum_{i=1}^L \Phi_i P_i \right) \underline{t}_0 \quad (65)$$

where  $P_i$  are partitioning matrices that have only elements on the main

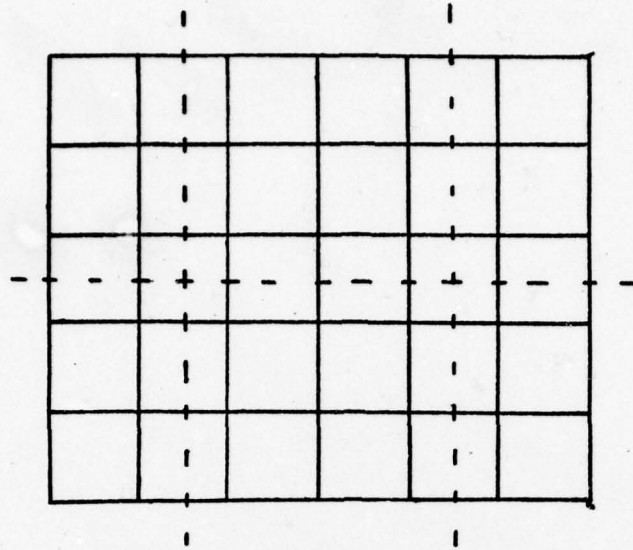


Figure 3. Coarse and Fine Meshes

diagonal and are required to satisfy

$$\underline{I} = \sum_{l=1}^L \underline{P}_l \quad (66)$$

where  $\underline{I}$  is the identity matrix.

Recalling the fine mesh matrix equation for the heat conduction problem

$$\underline{K} \underline{t} = \underline{q} \quad (56)$$

The exact solution will be approached after many iterations or

$$\underline{t}^\infty = \underline{K}^{-1} \underline{q} \quad (67)$$

The iterative solution is begun by considering a trial solution,  $\underline{t}^0$ , and then operating upon it to produce a new trial solution,  $\underline{t}^1$ .

$$\underline{t}^1 = \underline{B} \underline{t}^0 + \underline{z} \quad (68)$$

where  $\underline{B}$  is a matrix independent of iteration number

$\underline{z}$  is a vector independent of iteration number

Superscript denotes iteration number.

The error vector (Ref 7) is defined as

$$\underline{\epsilon}^n = \underline{t}^\infty - \underline{t}^n \quad (69)$$

and it is required that

$$0 = \lim_{n \rightarrow \infty} \underline{\epsilon}^n = \lim_{n \rightarrow \infty} (\underline{B})^n \underline{\epsilon}^0 \quad (70)$$

It is assumed that  $\underline{B}$  has a complete set of eigenvectors  $\underline{e}_i$  and corresponding eigenvalues,  $\lambda_i$ . The error may be expanded in terms of the eigenvectors.

$$\underline{\epsilon}^0 = \sum_i \alpha_i \hat{e}_i \quad (71)$$

where  $\alpha_i$  are expansion coefficients.

If a second iteration is done, then

$$\underline{\epsilon}^1 = \underline{B} \underline{\epsilon}^0 = \sum_i \alpha_i \lambda_i \hat{e}_i \quad (72)$$

Further iterations produce

$$\underline{\epsilon}^n = \sum_i \alpha_i \lambda_i^n \hat{e}_i \quad (73)$$

Stability of the iteration method requires that all the eigen-

values be less than one for the iteration process to converge.

$$\lim_{n \rightarrow \infty} \lambda_i^n = 0 \quad (74)$$

If the eigenvalues are ordered such that

$$1 > |\lambda_1| > |\lambda_2| > \dots > |\lambda_L| \quad (75)$$

then for a large number of iterations, the error is approximately (Ref 7)

$$\underline{\epsilon}^n \approx \alpha_i \lambda_i^n \hat{e}_i \quad (76)$$

The coarse mesh rebalancing method attempts to remove, or at least reduce, the effects of the second through the  $L^{\text{th}}$  eigenvalue, where there are  $L$  subregions in the coarse mesh. This makes equation (76) approximate (73) after fewer iterations. The coarse mesh solution has up to  $L$  eigenvectors that can be related to the eigenvectors of the fine mesh problem. The vector  $\underline{f}_1$  is defined by

$$\underline{f}_1 = \underline{P}_1 \underline{t}_0 \quad (77)$$

Because  $\underline{t}_0$  can be expanded in terms of its eigenvectors,  $\underline{f}_1$  can also be expanded in terms of these eigenvectors.

$$\underline{f}_1 = \sum_i \alpha_{i,1} \hat{e}_i \quad (78)$$

The eigenvectors are now classified as low or high eigenvectors.

Low eigenvectors are  $\hat{e}_i$ ,  $i=2,3,\dots,L$  where  $L$  is the number of subregions in the coarse mesh. The high eigenvectors are  $\hat{e}_i$ ,  $i>L$ . The dominant eigenvector  $\hat{e}_1$  is treated separately.

If the low eigenvectors were known and used as weighting vectors orthogonal to the error vector, then they would be eliminated when the rebalancing coefficients were determined (Ref 4).

The low eigenvectors are not known, but weighting vectors from the current iteration vector can be determined. The current iteration vector is an approximation for the solution, so eigenvectors from it make a logical approximation for the desired low eigenvectors. The use of this approximation can and does excite the high eigenvectors, while only suppressing the low eigenvectors.

The weighted residual method is used to obtain the arbitrary rebalancing coefficients. This method makes the residual orthogonal to independent weighting functions in an attempt to make the residual go to zero. The residual, the difference between two iterations, is used because the error is not known. The residual can be expressed as

$$\underline{r} = \underline{K} \left( \sum_i \underline{\Phi}_i \underline{P}_i \right) \underline{t}_0 - \underline{q} \quad (79)$$

When the residual is made orthogonal to the weighting vectors, the coarse mesh equation is obtained.

$$\sum_i \langle \underline{w}_k \underline{K} \underline{P}_i \underline{t}_0 \rangle \underline{\Phi}_i = \langle \underline{w}_k \underline{q} \rangle \quad (80)$$

where  $\langle \rangle$  indicates a scalar product.

This equation defines a set of  $L$  simultaneous equations that are solved for the undetermined multiplying coefficients.

Partitioning. There is one partitioning matrix for each subregion in the coarse mesh. The partitioning matrices are of size  $n \times n$ , where  $n$  is the number of nodal points in the fine mesh. The partitioning matrices are limited by

$$\sum_{l=1}^L P_l = \underline{I} \quad (81)$$

where  $\underline{I}$  is the identity matrix.

The simplest form of partitioning matrix, with this constraint, is either one or zero on the main diagonal depending upon whether the corresponding fine mesh point is in the coarse mesh subregion associated with that partitioning matrix.

Weighting Vectors. There exist many possible weighting vectors. The simplest to form and program is

$$W_l = P_l \underline{1} \quad (82)$$

where  $\underline{1}$  is a vector with all elements equal to one.

### III. Procedure

#### Computer

The Control Data Company (CDC) Cyber 7400 of the Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio was used exclusively during this study. All programming was done in the Fortran IV language.

#### Problem

The sample problem used to study acceleration, was a two-dimensional one centimeter square with a constant thermal conductivity of  $1.0 \text{ calories/cm}^3\text{-sec-}^\circ\text{C}$ . A uniform internal heat generation rate of  $144 \text{ calories/cm}^3\text{-sec}$  was specified. The left, top, bottom, and right boundaries were  $200$ ,  $300$ ,  $400$ , and  $500^\circ\text{C}$ , respectively.

A finite element grid as shown in Figure 4 was superimposed so that after elimination of the known boundary node temperatures, 1600 nodal points remained. Each nodal point had a corresponding linear simultaneous equation that was to be solved for the temperature at that point. The finite element mesh shown in Figure 4 was used to insure a coefficient matrix different from the coefficient matrix for finite differences. The use of simpler finite elements, which were isosceles right triangles (Fig 5), resulted in a set of equations that were identical to those obtained for finite differences (Ref 6: 285).

Because the finite difference method has already been studied, the conditions that provided the same set of equations were avoided, and a more general finite element was used. Finite difference equations and isosceles right finite elements result in five non-zero

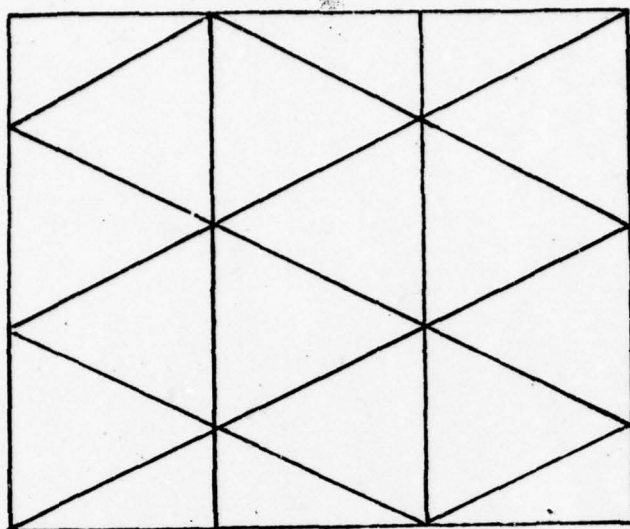


Figure 4. Finite Element Grid

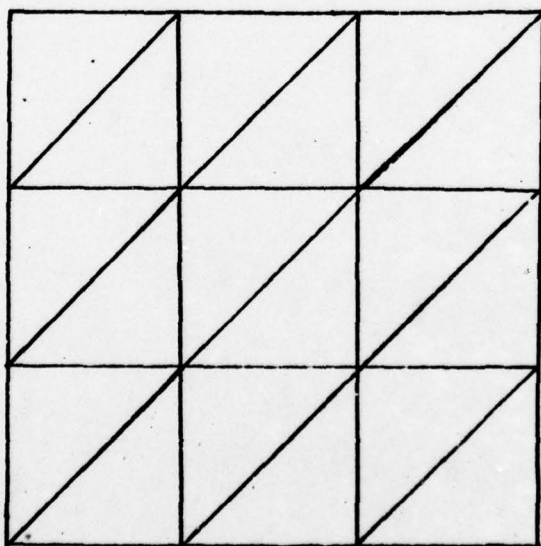


Figure 5. Isosceles Right Elements

elements in each row of the coefficient matrix. The more general finite elements have up to seven non-zero elements in each row of the coefficient matrix.

#### Programming

The solution of 1600 simultaneous equations, in general, requires a 1600 x 1600 matrix to be stored and manipulated during iterations. The matrix obtained for the finite element solution is a sparse matrix. There are no more than seven non-zero elements on each row of the matrix and with proper programming only non-zero elements need be stored.

Normalizing all main diagonal elements eliminates the need to store these values because the value one from the main diagonal can be contained in the programming rather than in memory. Normalizing is done by dividing all elements on a row and the corresponding element of the known vector by the value that would occur on the main diagonal for that row.

Because different scanning methods create coefficient matrices of different bandwidths, a bookkeeping system was created. This system stored the location and value of each of the non-zero off-diagonal elements in an integer and a real array, respectively. This resulted in two 6 x 1600 arrays or a total of 12 x 1600 values to store.

Symmetry was not utilized to further reduce the storage required because, although expected, symmetry could not be verified beforehand for all scans. The use of symmetry would have halved the storage required.

The reduction of the matrix to only non-zero elements also eliminates repeated multiplication by zero. Although these multiplications

would not affect the answer, they would require considerable computer execution time. The failure to take advantage of symmetry did not increase the number of multiplications. The symmetry occurs about the main diagonal, placing corresponding identical elements in the upper and lower triangular matrix parts, and the upper and lower matrices are treated differently by the iteration scheme (Equation 62).

#### Convergence Criteria

Iterations were continued until the temperature values at all nodal points had converged to within  $10^{-5}$  percent of the values of the last iteration. The convergence test is

$$\left| \frac{t^{n+1} - t^n}{t^{n+1}} \right| \leq 10^{-7} \quad (83)$$

where the superscript here denotes iteration number.

This test cannot be applied when one of the values is zero, because either a infinite or 100% value is obtained. When either the newly calculated or the previously calculated value is zero, the difference between these values is required to be  $10^{-7}$ .

$$|t^{n+1} - t^n| \leq 10^{-7} \quad (84)$$

#### Trial Vector

The initial trial vector for successive overrelaxation must be supplied. The vector supplied had all elements equal to one. This vector, as well as other vectors with no zero elements, does not

require the use of the special convergence criteria unless the solution actually converges to a value near zero. This has been identified because a vector with all elements set to zero is a common choice for an initial trial vector (Ref 1, 4).

The initial trial vector also has an advantage when used with coarse mesh rebalancing. Because the rebalancing factors will approach one as the problem converges and will be near one at all times, starting at one can reduce convergence time.

### Scanning

Three scans were used in this study. They are the same as scans used by Pearson (Ref 1) and Cudahy (Ref 2).

Serial Scan. The serial scan is shown in Figure 6. The associated matrix is shown in Figure 7. This scan does not take advantage of working close to a known boundary early. This is a simple scan and is probably the most common. Points are numbered from bottom to top in columns from left to right. Serial scanning has the advantages of simple computer programming and the narrowest bandwidth. The narrow bandwidth can save computer storage, if the entire matrix were being used. Only the non-zero band would have to be stored. The difference in bandwidth does not affect the program used for these solutions because it was designed out of the program. Bandwidth was designed out because scans with wide bandwidths were to be utilized.

Modified Serial Scan. The modified serial scan is shown in Figure 8. The associated matrix is shown in Figure 9. This scan takes advantage of solving for values near both the left and right

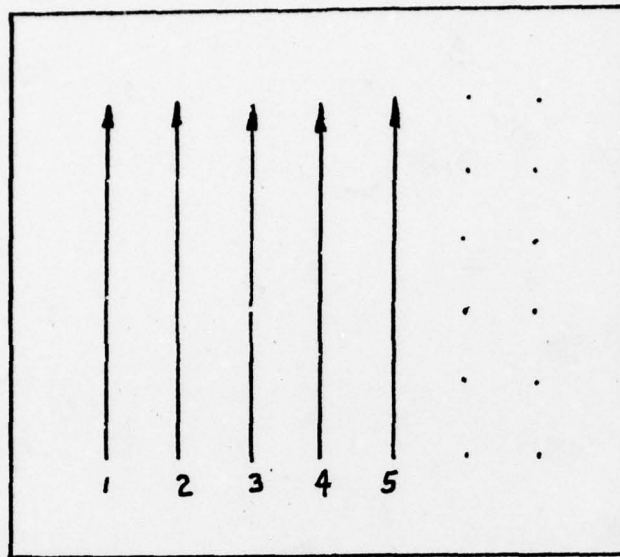


Figure 6. Serial Scanning

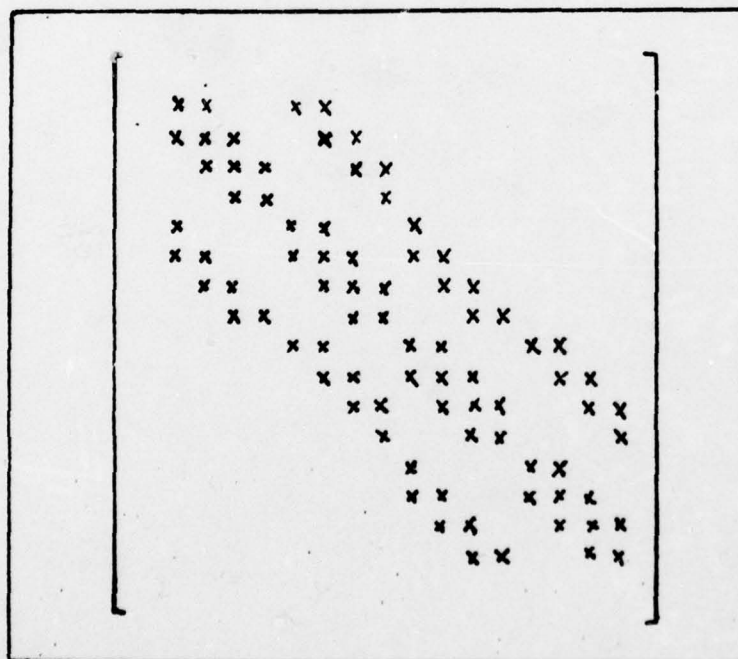


Figure 7. Matrix for Serial Scan

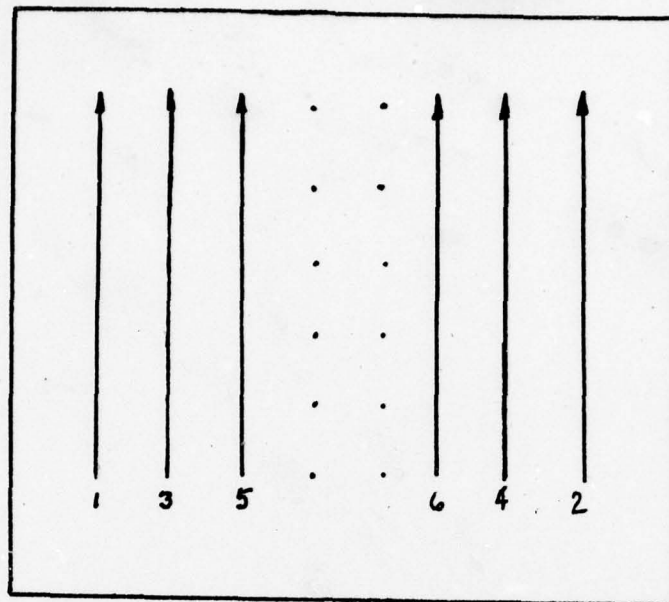


Figure 8. Modified Serial Scanning

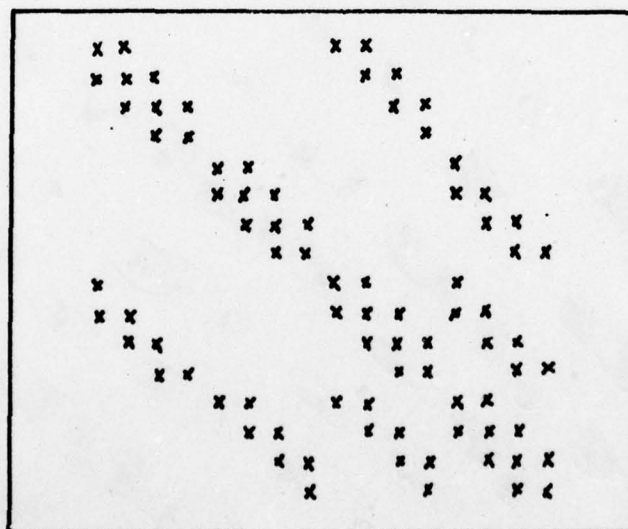


Figure 9. Matrix for Mod-Serial Scan

boundaries early. The points are scanned from bottom to top, as in the serial scan, but first on the left then the right and so on. This scan requires programming logic that performs the necessary jumps from left to right and back again. The bandwidth of the matrix for this scan is wider than for the serial scan.

Optical Scan. The spiral scan is shown in Figure 10. The associated matrix is shown in Figure 11. This scan has the broadest bandwidth of the three scans. This scan also requires the most programming to negotiate the corners to number the mesh points. This scan attempts to take advantage of known values as soon as possible to speed convergence.

#### Coarse Mesh Rebalancing

Coarse mesh rebalancing should have an accelerating effect dependent upon the successive overrelaxation factor, the size of the coarse mesh, and the frequency with which the iteration vector is rebalanced. These parameters were varied to determine their effects upon convergence rates.

The coarse mesh rebalancing method was applied only to the serial scan.

The 1600 (40 x 40) node points were divided into 25 (5 x 5), 36 (6 x 6), 49 (7 x 7), and 64 (8 x 8) coarse mesh subregions. These were selected to test the recommendation (Ref 9:275) that the number of coarse mesh regions should be about the square root of the number of node points.

The coarse mesh rebalancing method requires the solution of a matrix equation of order equal to the number of subregions in the

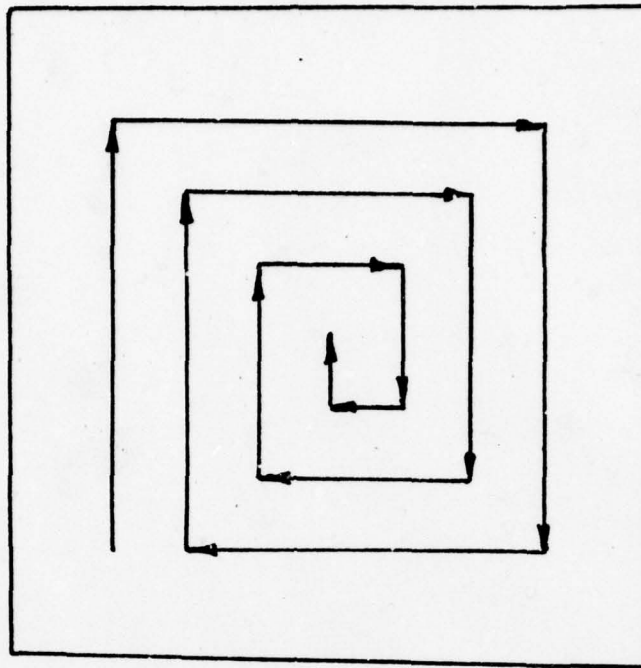


Figure 10. Spiral Scanning

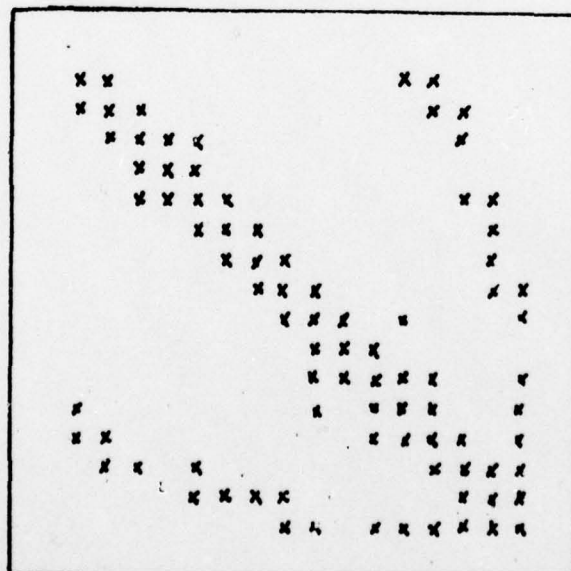


Figure 11. Matrix for Spiral Scan

coarse mesh. The additional time to solve the larger matrix problems for rebalancing can offset the reduction in the number of iterations that may occur.

The coarse mesh rebalancing method consists of the following:

1. Dividing the region into subregions with the coarse mesh and assigning each nodal point to the appropriate sub-region.
2. Interrupt the iteration scheme after a set number of iterations.
3. Create a matrix equation from equation (80).
4. Solve this matrix equation for the rebalancing coefficients.
5. Multiply each term in the iteration vector by the appropriate rebalancing factor.
6. Use the rebalanced vector in the iteration scheme, returning to step 2 until convergence occurs.

A simple example is contained in Appendix A.

The iteration vector was rebalanced after 3, 4, 5, 8, 10, and 15 iterations to determine the effect of rebalancing frequency on the number of iterations and computer execution time.

The successive overrelaxation factor was varied to determine its effect on convergence rate. It was predicted (Ref 4) that the use of the optimum overrelaxation factor would negate the effects of rebalancing. The optimum and four other overrelaxation factors were used to test this and determine other effects of varying this parameter.

The matrix equation for rebalancing coefficients was solved using successive overrelaxation. The overrelaxation factor was fixed at 1.50 and the same convergence criteria as applied to the total

problem was used. This allowed the use of the same successive over-relaxation subroutine, limiting the additional programming for coarse mesh rebalancing.

#### IV. Results

##### Scanning

Table I contains the number of iterations and computer execution time required for the finite element solution of the test problem by the three scanning methods with the optimum overrelaxation factors.

Fewer iterations were required for the modified serial scan than for the serial scan. Fewer iterations were required for the spiral scan than for the modified serial scan. Scanning near known boundaries early does reduce the number of iterations required for convergence.

The computer execution time listed in Table I is for execution of the program only and does not include program compilation time, input, or output times. The computer execution times have exactly the opposite trend that the numbers of iterations have with scanning. More execution time is required for the scans that scan near the known boundaries early. This is believed to be due to the extra computer programming needed to number the node points, and determine which node points are vertices of each element. The scans which scan the boundaries early, require logic that must negotiate jumps and corners (Fig 8 and 10). The difference in programming is also reflected in the different program compilation times.

The optimum overrelaxation factors were determined by repeatedly solving the problem with different overrelaxation factors and comparing the number of iterations. Figures 12, 13, and 14 show the number of iterations for convergence versus overrelaxation factor

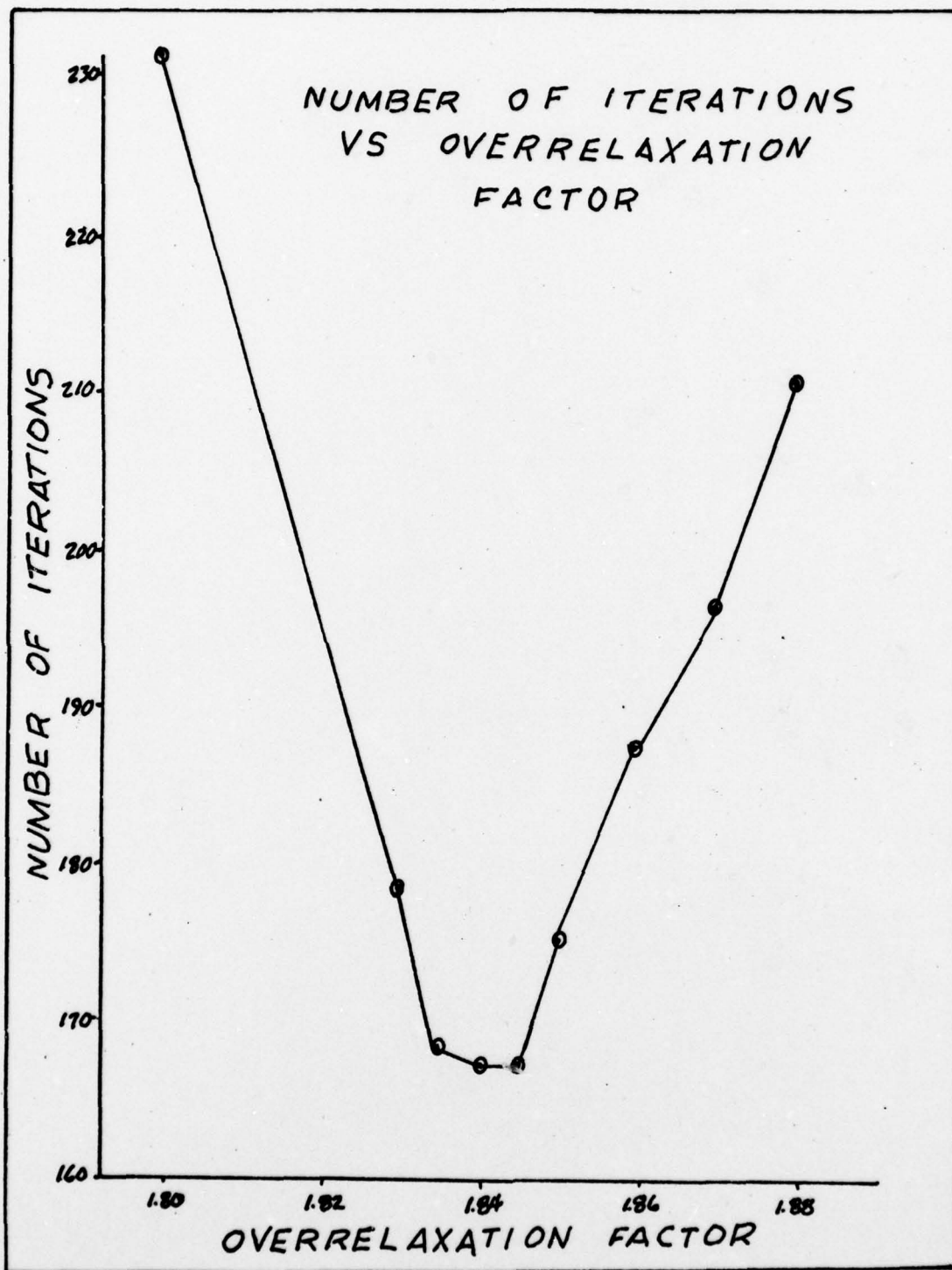


Figure 12. Graphic Determination of Optimum Overrelaxation Factor for the Serial Scan

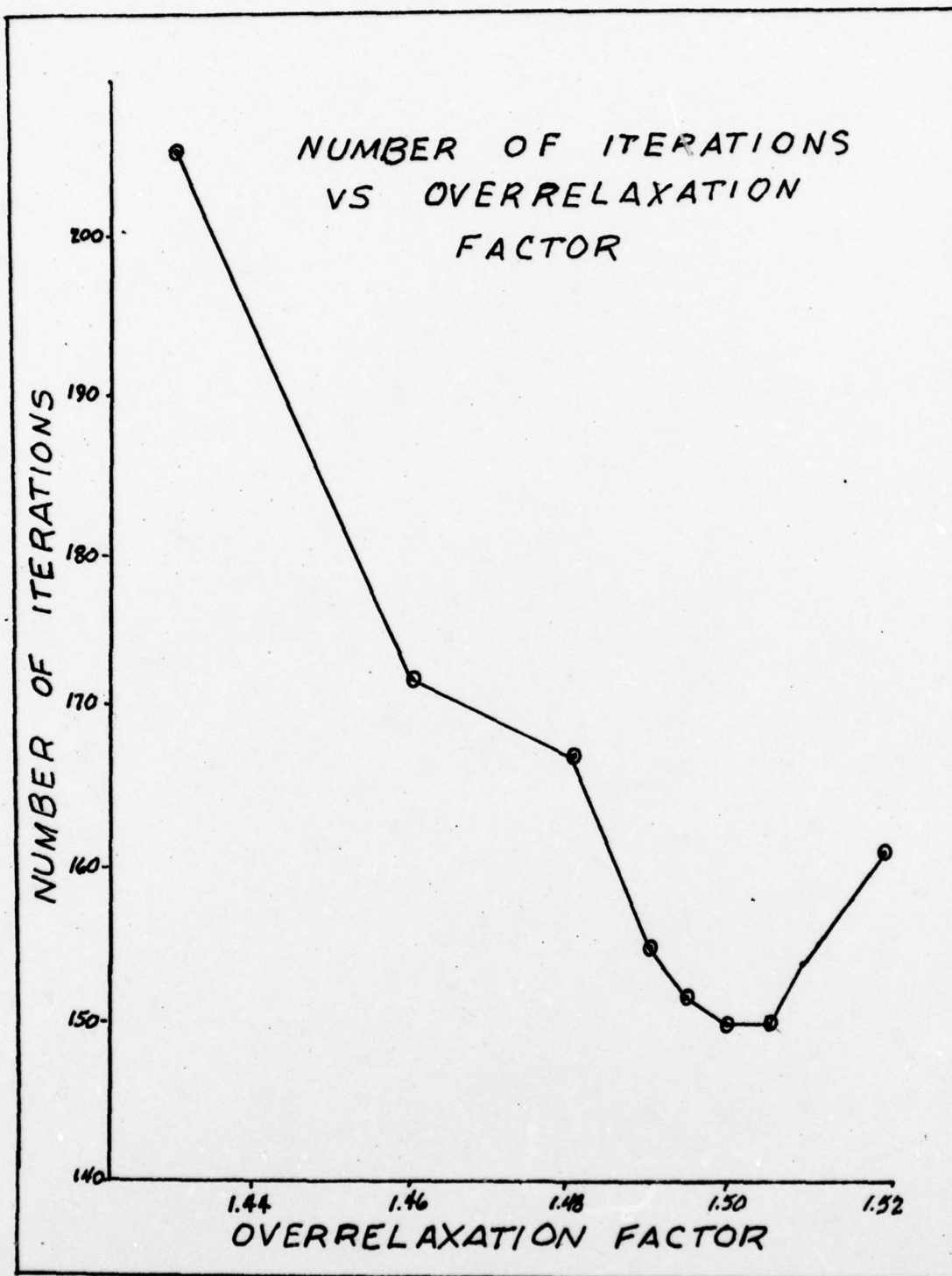


Figure 13. Graphic Determination of Optimum Overrelaxation Factor for the Modified Serial Scan

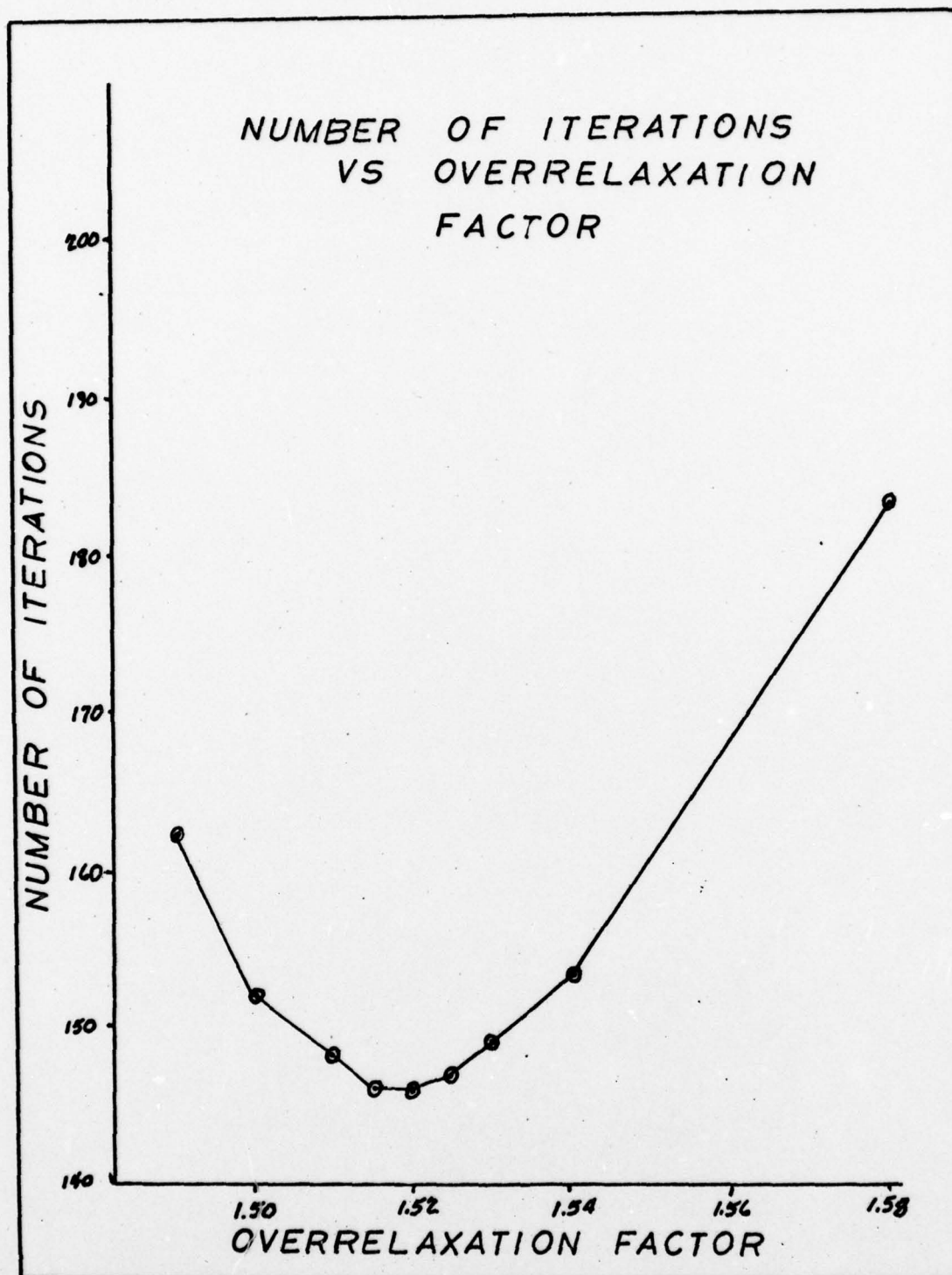


Figure 14. Graphic Determination of Optimum Overrelaxation Factor for the Spiral Scan

Table I  
Scanning Results

Scan	Overrelaxation Factor (Optimum)*	Number of Iterations	Computer Execution Time (sec)**	Program Compilation Time (sec)
Serial	1.84	167	17.80	0.98
Modified Serial	1.50	150	18.56	1.26
Spiral	1.52	146	18.96	1.46

\* Determined experimentally

\*\* Does not include compilation nor input and output times.

near the optimum overrelaxation factors for the three scans tested. The determination of optimum values was limited by the leveling off of the number of iterations over a range of overrelaxation factors. This was not considered to be a problem because the solution was obtained in the same number of iterations for the specified convergence criteria. Any value within this range could be considered optimum for the sample solution.

These results for the finite element solution when compared to the results of Pearson (Ref 1:29-33) for the finite difference solution show a number of similarities. Pearson had a similar reduction in the number of iterations, about 15% for scanning near known boundaries rather than serial scanning. This study had 10% fewer iterations for the modified serial scan and 13% fewer iterations for the spiral scan when compared to the serial scan. Pearson required

fewer iterations for the modified serial scan than for the spiral scan for three of his four problems. Pearson's fourth problem required fewer iterations for the spiral scan than for the modified serial scan. This study required fewer iterations for the spiral scan than for the modified serial scan. The problems Pearson used each had different boundary conditions; this study had boundary conditions different from any of those used by Pearson. The boundary scan, that is faster for a particular problem, might be determined by the type, magnitudes, and ratios of the boundary conditions.

Pearson experienced a 200% to 300% increase in computer execution time for the boundary scans. This study had a slight increase in the computer execution time for the boundary scans. The exact method of storage and manipulation of the matrix equations used by Pearson is not known, and no conclusions can be drawn except that he attributes the increase to the same factors.

#### Coarse Mesh Results

Coarse mesh rebalancing was applied to the serial scan solution to the problem and the results are presented in Table 2 through 6.

Table 2 contains the results from solutions using an overrelaxation factor of 1.50. The solution without rebalancing required many more iterations and more computer execution time than the rebalanced solutions. In all cases with an overrelaxation factor of 1.50, the coarse meshes with the most subregions (64) required fewer iterations than any other coarse mesh solution with the same rebalancing frequency. Those solutions with 36 subregions in the coarse required the least computer execution time for each rebalancing

Table II

Coarse Mesh Results; Overrelaxation Factor = 1.50

Coarse mesh Subregions	Iterations Between Rebalancing	Number of Iterations	Number of Rebalancing	Total Execution Time (sec)
--	NA	689	0	73.05
25	3	126	41	20.56
36	3	111	36	19.90
49	3	102	33	22.65
64	3	90	29	27.58
25	5	170	33	23.60
36	5	153	30	22.97
49	5	139	26	24.85
64	5	133	26	30.62
25	8	228	28	29.52
36	8	206	25	28.02
49	8	191	23	29.24
64	8	180	22	33.92
25	10	255	25	31.78
36	10	234	23	30.66
49	10	219	21	31.87
64	10	206	20	35.65
25	15	319	21	38.51
36	15	293	19	35.78
49	15	275	18	37.32
64	15	264	17	40.63

frequency. Solutions with the same coarse mesh size required fewest iterations and least computer execution time when rebalanced after every three iterations. This was the highest rebalancing frequency tested. The fastest solution with an overrelaxation factor of 1.50 was the solution with the coarse mesh of 36 subregions and rebalancing after every three iterations.

Table III contains the results for solutions using an overrelaxation factor of 1.60. The same results that were observed for an

Table III

Coarse Mesh Results; Overrelaxation Factor = 1.60

Coarse mesh Subregions	Iterations Between Rebalancing	Number of Iterations	Number of Rebalancing	Total Execution Time (sec)
NA	NA	525	0	56.80
25	3	114	37	18.63
36	3	102	33	18.66
49	3	99	32	21.99
64	3	84	27	26.39
25	4	136	33	20.45
36	4	124	30	20.34
49	4	111	27	21.94
64	4	103	25	26.99
25	5	154	30	21.79
36	5	139	27	21.29
49	5	133	26	23.61
64	5	129	25	30.00
25	6	172	28	23.39
36	6	160	26	23.32
49	6	149	24	25.08
64	6	142	23	29.89
25	8	204	25	25.81
36	8	188	23	25.32
49	8	174	21	26.66
64	8	166	20	31.17
25	10	228	22	29.01
36	10	214	21	28.69
49	10	197	19	29.22
64	10	189	18	33.14

overrelaxation factor of 1.50 were observed with one exception. The solution with a rebalancing frequency of three iterations and the coarse mesh with 25 subregions was the fastest. The solution with the coarse mesh of 36 subregions was only 0.03 seconds slower, about 0.2%

Table IV contains the results for solutions using an overrelax-

Table IV

Coarse Mesh Results; Overrelaxation Factor = 1.70

Coarse mesh Subregions	Iterations Between Rebalancings	Number of Iterations	Number of Rebalancing	Total Execution Time (sec)
NA	NA	376	0	40.81
25	3	102	33	16.66
36	3	96	31	17.30
49	3	98	32	21.25
64	3	83	27	24.68
25	4	123	30	18.57
36	4	111	27	18.18
49	4	103	25	19.90
64	4	95	23	24.63
25	5	134	26	18.85
36	5	123	24	19.06
49	5	123	24	22.12
64	5	123	24	27.94
25	8	172	21	22.31
36	8	160	20	21.86
49	8	151	18	23.31
64	8	150	18	28.33
25	10	194	19	24.47
36	10	184	18	24.08
49	10	175	17	25.27
64	10	166	16	28.72
25	15	231	15	27.79
36	15	217	14	27.01
49	15	207	13	27.69
64	15	202	13	31.06

ation factor of 1.70. Trends similar to those observed for overrelaxation factors of 1.50 and 1.60 were again obtained. The coarse mesh with 25 subregions was faster than the other coarse meshes for rebalancing frequencies of three and five iterations, but not four iterations. The coarse mesh with 36 subregions was still the fastest of those solutions with a rebalancing frequency of 4 iterations. The

Table V

Coarse Mesh Results; Overrelaxation Factor = 1.80

Coarse mesh Subregions	Iterations Between Rebalancings	Number of Iterations	Number of Rebalancing	Total Execution Time (sec)
NA	NA	231	0	25.04
25	3	108	35	17.37
36	3	112	37	19.64
49	3	113	37	25.57
64	3	107	35	28.68
25	4	127	31	19.45
36	4	114	28	18.83
49	4	111	27	21.70
64	4	107	26	26.43
25	5	125	24	17.67
36	5	119	23	17.83
49	5	122	24	21.11
64	5	128	25	27.48
25	8	139	17	18.06
36	8	131	16	17.70
49	8	127	15	19.23
64	8	126	15	23.33
25	10	142	14	17.43
36	10	135	13	17.24
49	10	136	13	19.34
64	10	134	13	22.58
25	15	165	10	19.55
36	15	156	10	19.00
49	15	156	10	20.32
64	15	147	9	21.67

fastest solution with an overrelaxation factor of 1.70 was the solution with a rebalancing frequency of three iterations and the coarse mesh with 25 subregions.

The results of those solutions done with an overrelaxation factor of 1.80 are presented in Table V. The results with this overrelaxation

factor differed from the previously discussed results in a number of ways. The number of iterations required was not inversely proportional to the number of subregions in the coarse mesh for all rebalancing frequencies. The fastest solution was obtained with a rebalancing frequency of 10 iterations, much lower than for the other overrelaxation factors. One solution with the high rebalancing frequency (3 iterations) was almost as fast as fastest. The occurrence of fast solutions at lower rebalancing frequencies is believed due to the proximity of the overrelaxation factor to the optimum of 1.84.

Table VI contains the results from solutions with an overrelaxation factor of 1.84, the optimum. The solution without rebalancing required less time than all but four of the solutions with this overrelaxation factor occur at the slower frequencies. The fastest solution occurred with a rebalancing frequency of ten iterations and the second fastest at a frequency of eight iterations. The number of iterations required did not drop as rapidly as with other overrelaxation factors when the coarse mesh had more subregions.

Solutions faster than the unbalanced solution with the optimum overrelaxation factor occurred with overrelaxation factors of 1.70, 1.80, and 1.84. A solution with an overrelaxation factor of 1.50 was within 11% of the unbalanced solution with the optimum overrelaxation factor. A solution with an overrelaxation factor of 1.60 was within 5% of the time for solution with the optimum overrelaxation factor and no rebalancing. These results indicate that a problem can be significantly accelerated even if the optimum overrelaxation factor is not known. These solution times also indicate that the time of solution

Table VI

Coarse Mesh Results; Overrelaxation Factor = 1.84

Coarse mesh Subregions	Iterations Between Rebalancings	Number of Iterations	Number of Rebalancing	Total Execution Time (sec)
NA	NA	164	0	17.80
25	3	134	44	21.44
36	3	122	40	21.38
49	3	132	43	27.27
64	3	125	41	33.27
25	4	142	35	21.06
36	4	124	31	19.90
49	4	127	31	23.93
64	4	132	33	30.46
25	5	127	25	17.62
36	5	128	25	18.80
49	5	127	25	21.30
64	5	144	28	30.25
25	8	148	18	18.94
36	8	129	16	17.38
49	8	126	15	18.71
64	8	123	15	21.92
25	10	133	13	16.73
36	10	133	13	17.45
49	10	131	13	18.71
64	10	137	13	23.07

can approach or surpass the unrebanded solution with the optimum overrelaxation factor.

If the optimum overrelaxation factor is known, it will solve the problem almost as quickly as the best rebanded solution. The best solution required 16.66 seconds. The unrebanded solution with the optimum overrelaxation factor took 17.80 seconds. This is about a 7% increase. Because the rebalancing frequency and coarse

mesh size which give the fastest solution cannot be predicted, solving with the optimum overrelaxation factor, if known, is a good strategy.

In general, the suggestion of Wachspress (Ref 9:275), that the number of subregions in the coarse mesh be about the square root of the number of node points, is borne out. In this case, the coarse mesh with 36 subregions was closest to the square root of the number of fine mesh points. This size coarse mesh usually provided the fastest solution of all the different coarse meshes. This size coarse mesh provided the second fastest solution if not the fastest. When the optimum overrelaxation factor is not known, solving using rebalancing and a coarse mesh with about as many subregions as the square root of the number of mesh points would be part of a good strategy for solution.

The rebalancing frequency which gave the fastest solution, varied with overrelaxation factor. The frequency that gave the fastest results was three iterations, when not near the optimum overrelaxation factor. As for those solutions, when near but not at the optimum overrelaxation factor, three iterations still give fast solution. When at the optimum overrelaxation factor, a lower rebalancing frequency gives a faster solution than the three iteration frequency. Estimating the optimum overrelaxation factor will make the choice of a rebalancing frequency easier. When near the optimum overrelaxation factor, a frequency of eight or ten iterations would be a good strategy. When a good estimate of the optimum overrelaxation factor cannot be made, as in most cases, a high frequency, about three would be part of a good strategy for a fast solution.

The time to solve the larger system of equations for the rebalancing coefficients, when the coarse mesh has more subregions, must

be weighed against the reduction in the number of iterations. The effect of the size of the coarse mesh on the time needed to rebalance can be seen in the solutions done with an overrelaxation factor of 1.84 and a rebalancing frequency of 10 iterations. The same number of rebalancings was done and nearly the same number of iterations was done, but the time for solution increased with the size of the coarse mesh.

The maximum error vs the number of iterations was plotted for all solutions using rebalancing. Selected plots are presented in Figures 15 through 22; the remaining plots are in Appendix B. The unbalanced solution is plotted with each family of curves for comparison.

The unbalanced solutions all approach a straight line on the semi-logarithmic plots. The rebalanced solutions all show induced errors each time a rebalancing is done. This induced error along with the iteration scheme error are reduced quickly after rebalancing. This effect gives the spikes, which are especially apparent in Figure 15. For those solutions with low rebalancing frequencies (Fig 15 and 17), the maximum error approaches a straight line in between rebalancing spikes. Those solutions which are rebalanced more frequently (Fig 16 and 18), do not show the straight line, and have smaller rebalancing spikes.

These effects are due to excitation and suppression of the eigenvector components by rebalancing. When the system is rebalanced, high frequency eigenvector component are excited and low frequency eigenvector components are suppressed. The excitation causes the increase in error, and the suppression of low frequency eigenvector components

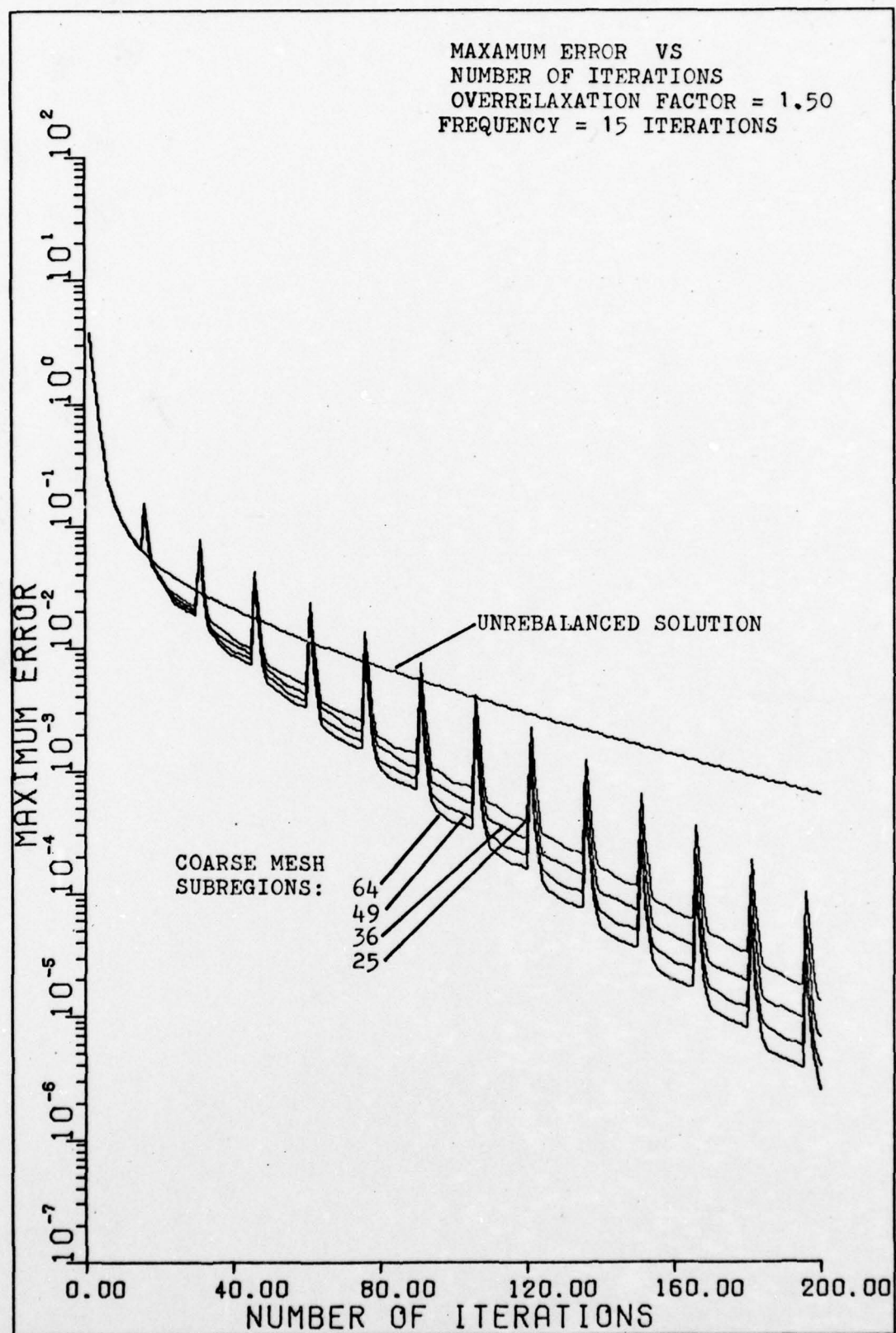


Figure 15. Effect of Rebalancing

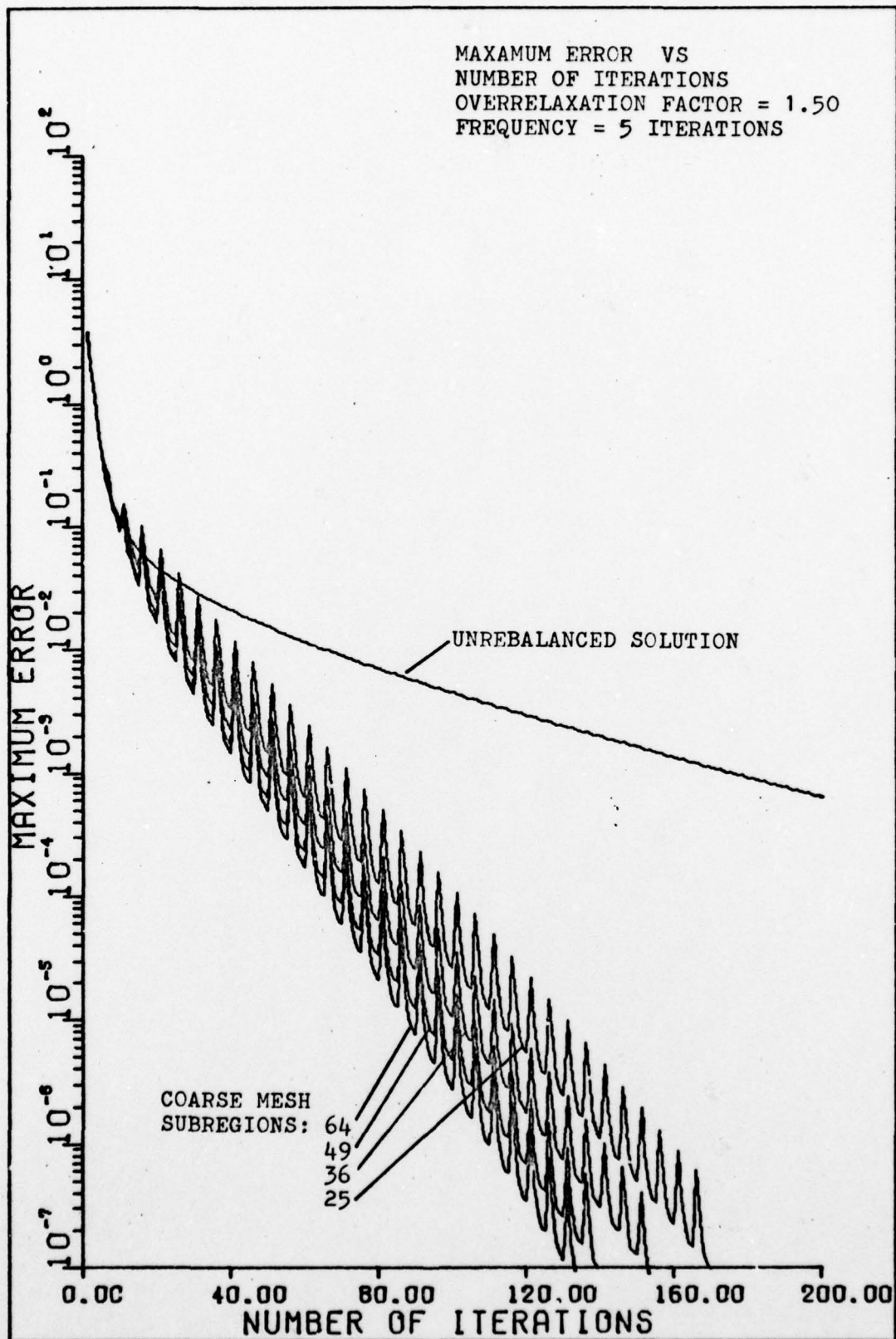


Figure 16. Effect of Rebalancing

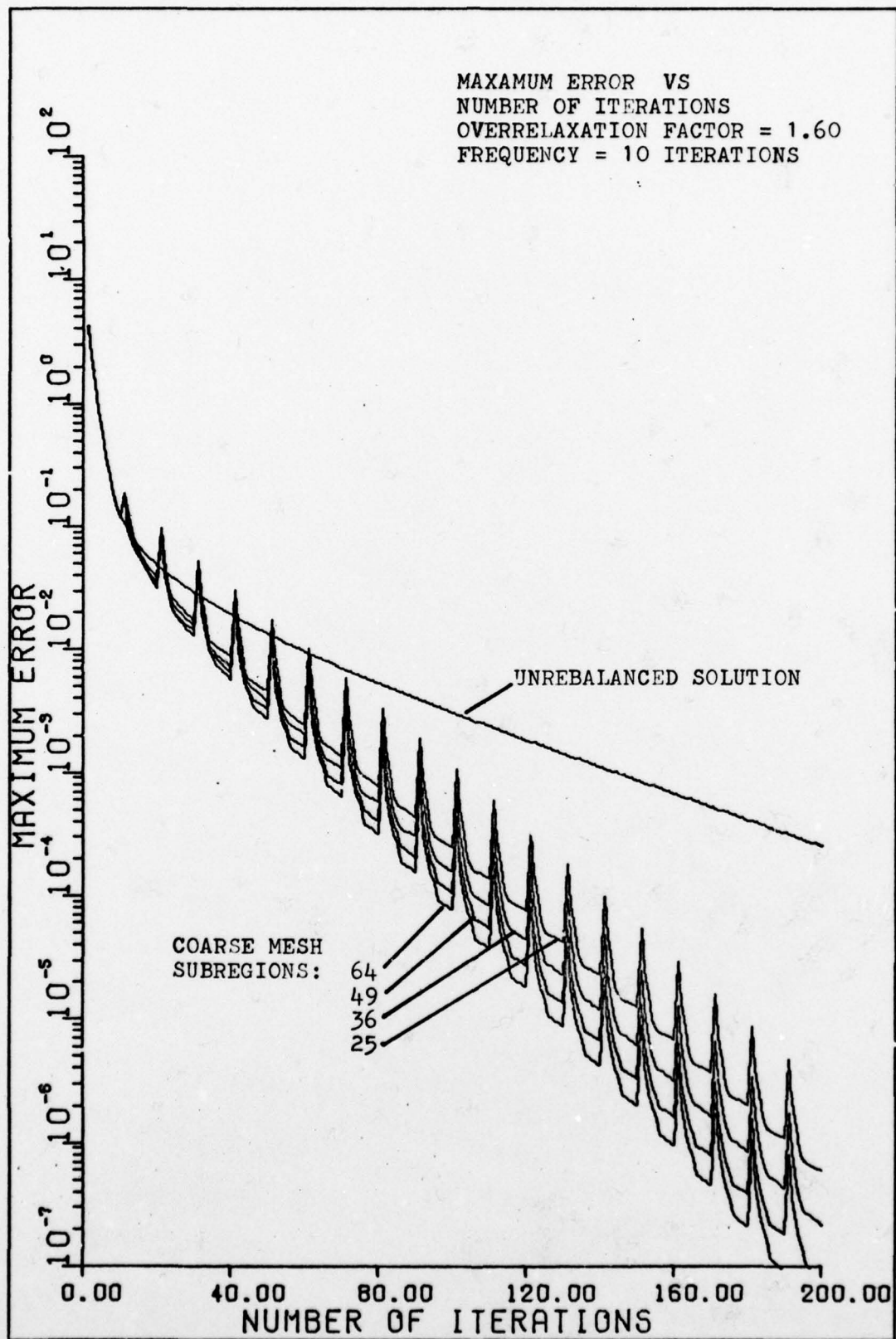


Figure 17. Effect of Rebalancing

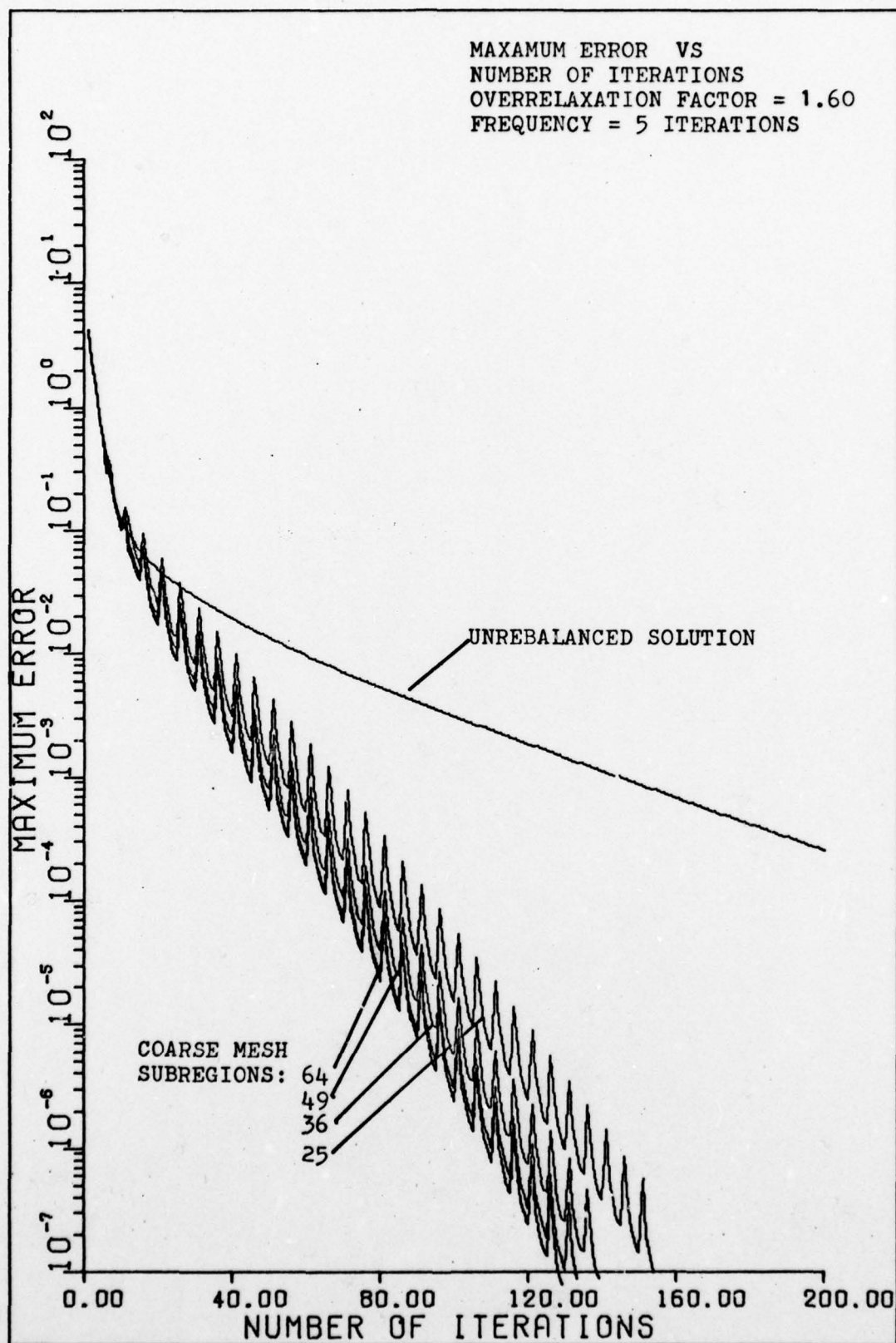


Figure 18. Effect of Rebalancing

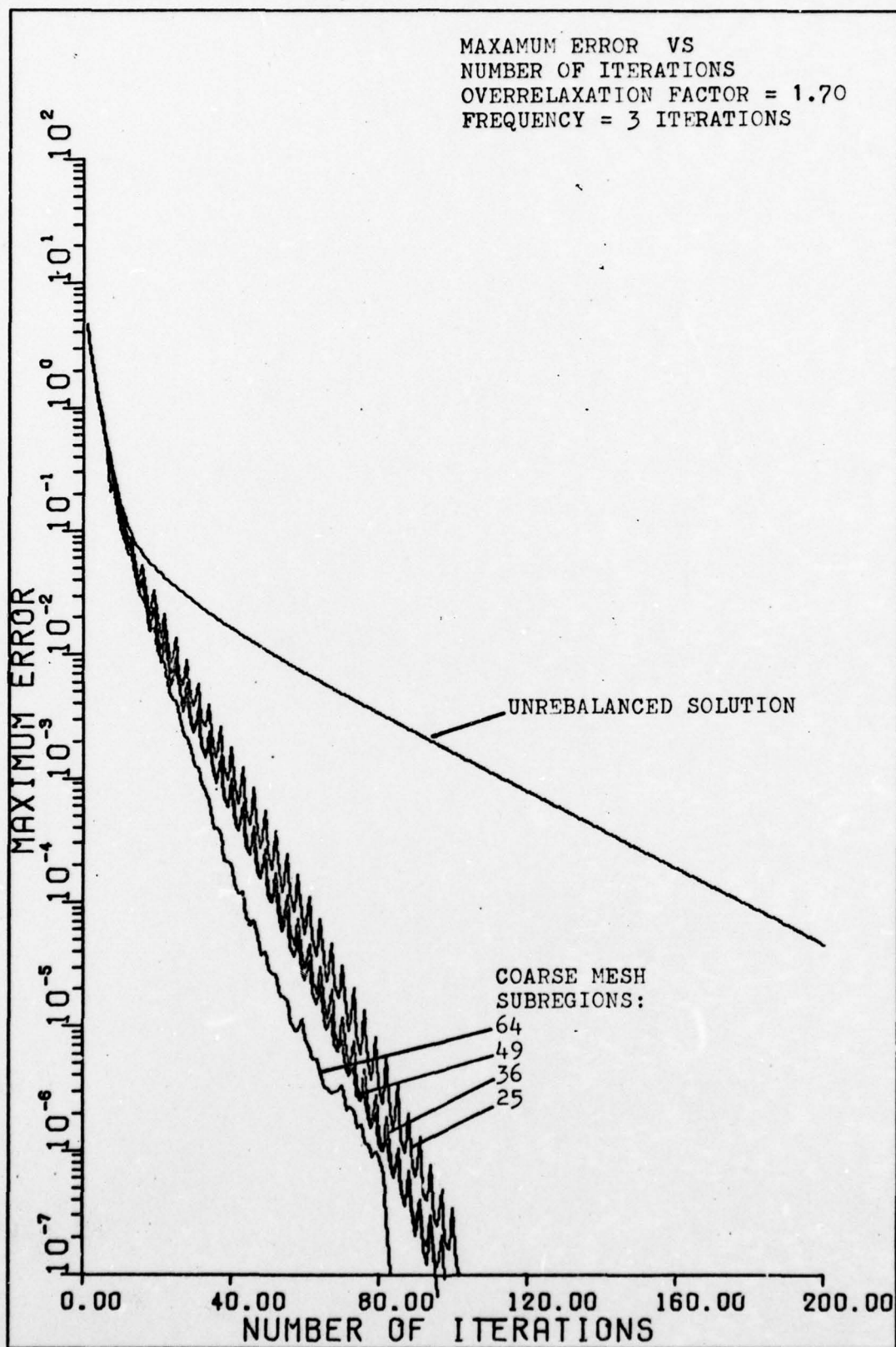


Figure 19. Effect of Rebalancing

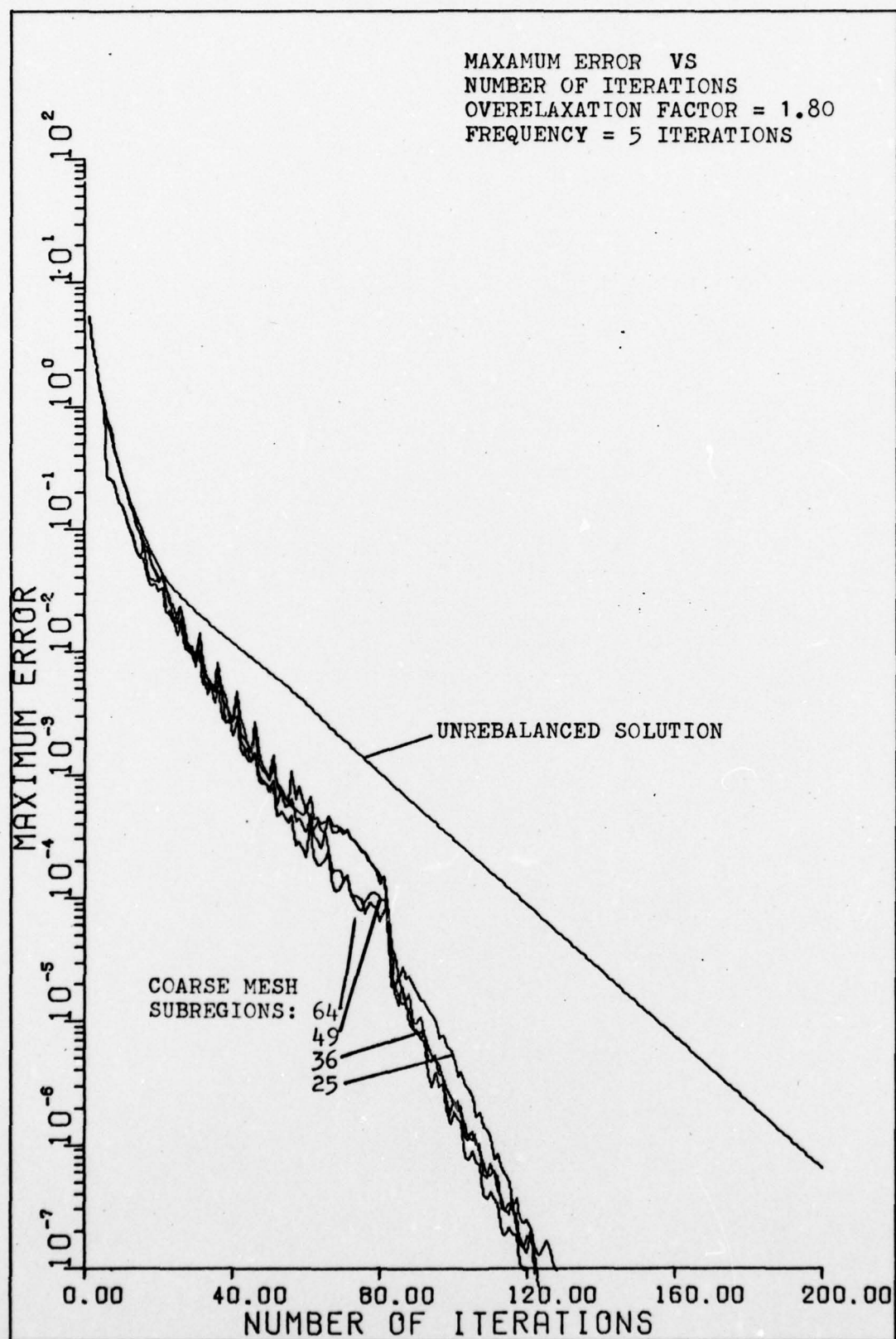


Figure 20. Effect of rebalancing

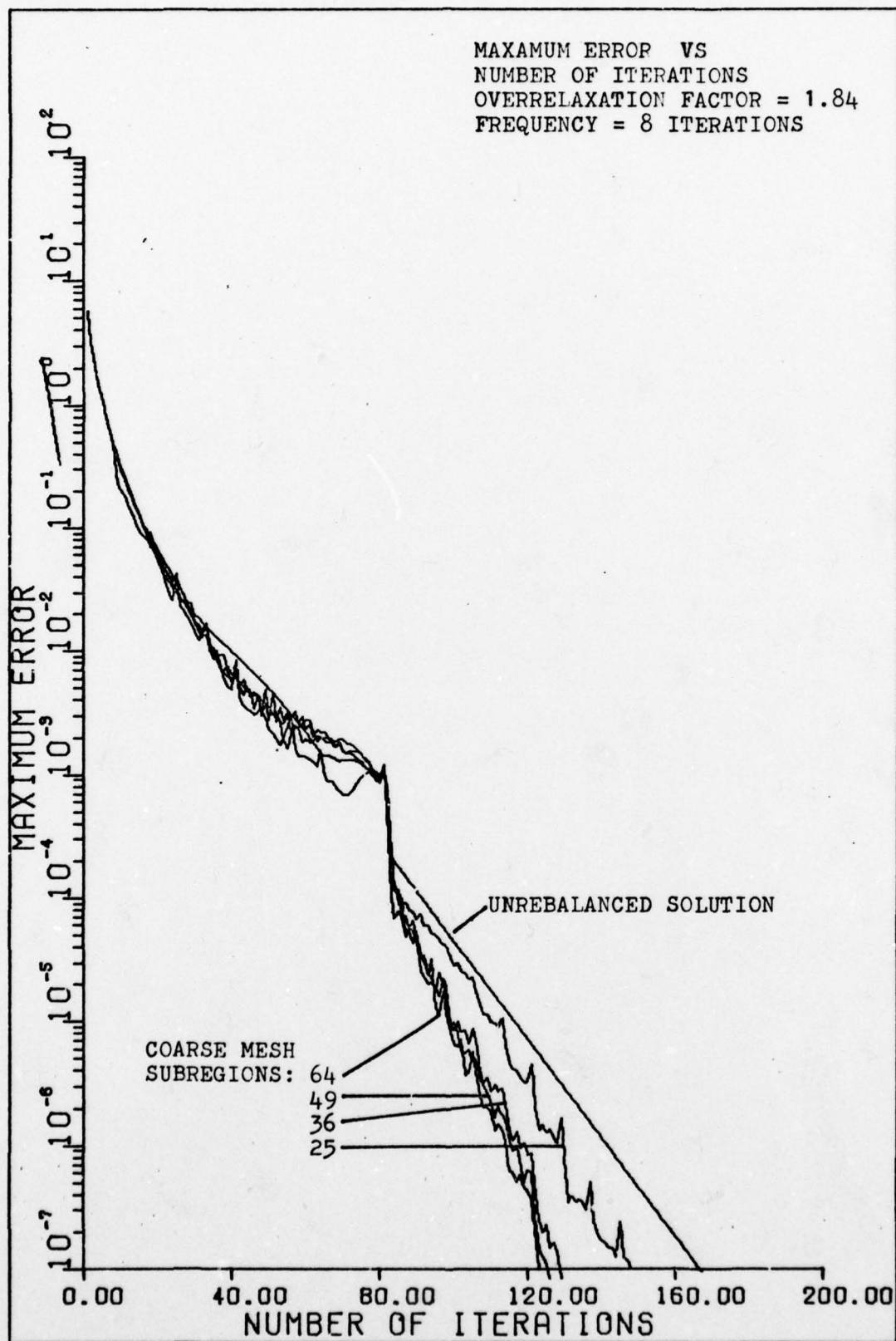


Figure 21. Effect of Rebalancing

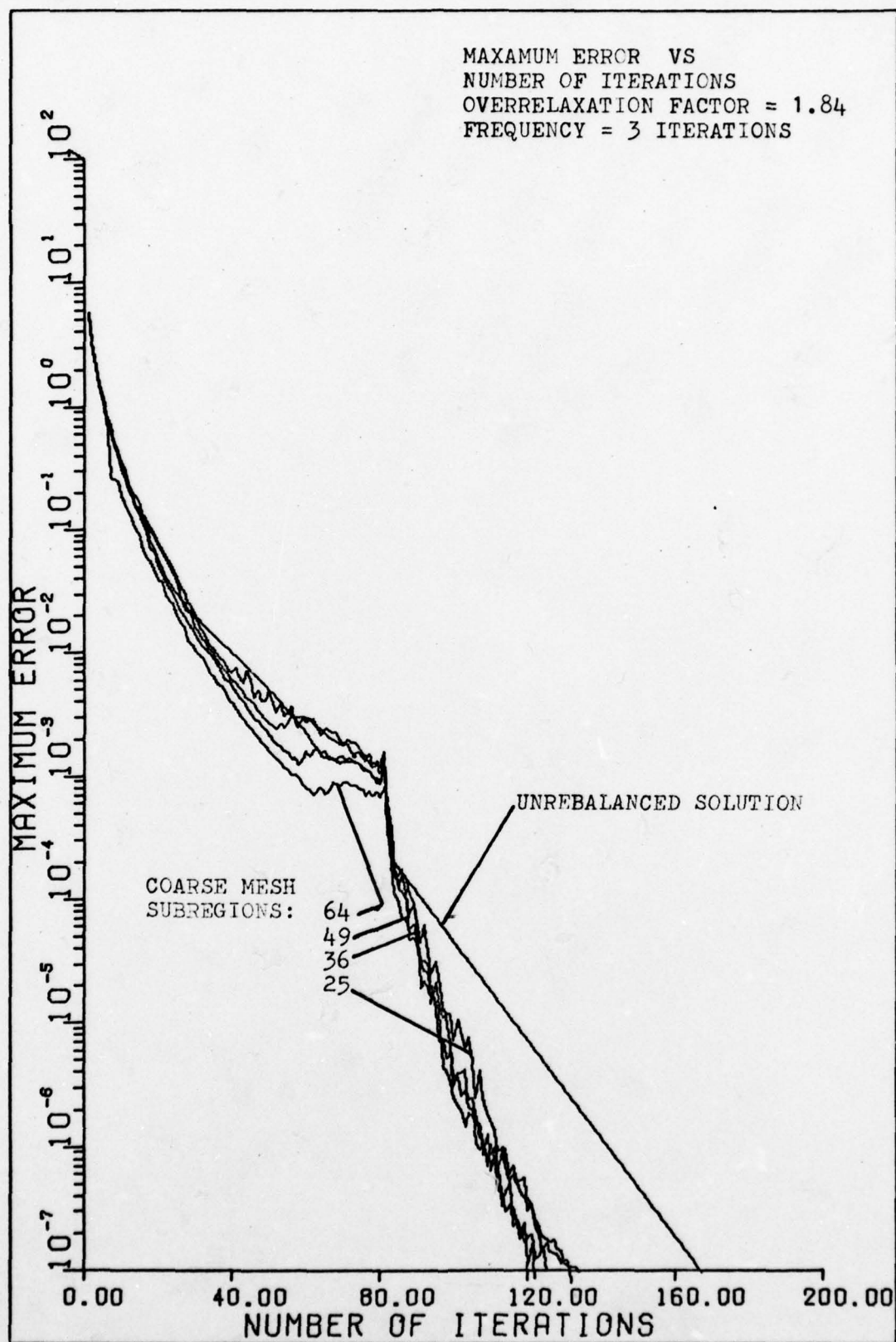


Figure 22. Effect of Rebalancing

causes the rapid decrease in the error. The normal iteration process allows the suppressed components to return after a number of iterations. This causes the curves to approach a straight line again.

The rebalancing frequencies which are more successful are those which rebalance before the straight line is apparent.

The error induced is lower when rebalancing is done more often. This is due to lingering suppression of some of the eigenvectors that might otherwise be excited.

An interesting effect occurs during the solution using the optimum overrelaxation factor and no rebalancing (Fig 21). The error in this case reaches a point where it is not reduced by further iterations for a period and then is reduced very rapidly. This also occurs with the rebalanced solutions at the same point. A similar effect occurs for the rebalanced solutions with an overrelaxation factor of 1.80. This behavior may be due to an eigenvector excitation and suppression due to the optimum or near optimum overrelaxation factors. The use of the optimum and near optimum reduces the accelerating effect of coarse mesh rebalancing. This can be seen when comparing Figures 20-22 with Figures 15-19. The curves with the higher overrelaxation factors have virtually no spikes in some cases and the curves are close together. The solutions with nonoptimum overrelaxation factors have distinct spikes and show an effect from different coarse mesh sizes.

Nakamura (Ref 4) predicted that use of the optimum overrelaxation factor would destroy the effects of rebalancing. This was found to be true in part; the accelerating effect was reduced but was still present when the optimum overrelaxation factor was used. In addition, this study showed a change in the best rebalancing frequency with overrelaxation factor.

## V. Conclusions and Recommendations

### Conclusions

From the results of the three different scanning techniques, it is concluded that the scanning of known boundaries first and the subsequent ordering of the simultaneous equations does transfer boundary effects into the mesh at a rapid rate. The scan which produced the greatest reduction in the number of iterations was the spiral scan. The two boundary scans tested both required more computer execution time than the classic serial scan, even though fewer iterations were required. The number of iterations alone is not an accurate measure of the efficiency of a solution method.

The solution of a problem using coarse mesh rebalancing and arbitrary overrelaxation factors can result in solutions that required computer execution times similar to the solution with the optimum overrelaxation factor. This precludes the necessity of finding or approximating the optimum overrelaxation factor. If, however, the optimum overrelaxation factor is known, it should be used for the solution and the iteration scheme not rebalanced. This conclusion is made because little improvement is possible over the time with the optimum overrelaxation factor, the factors which would improve the time for solution cannot be predicted accurately, and negative effects are very likely with rebalancing and the optimum overrelaxation factor.

The size of the coarse mesh, when coarse mesh rebalancing is applied, should be sized to have about as many subregions as the square root of the number of node points in the fine mesh.

### Recommendations

The plots of maximum error versus number of iterations show a steep slope immediately after rebalancing, and a lesser slope as iterations continue. Monitoring this slope might provide a method of determining the best point to rebalance rather than the constant frequency method used in this study. This would be most advantageous when a large number of coarse mesh subregions were required.

This study was limited to one set of weighting vectors and one set of partitioning matrices. Many other combinations are possible, and could be explored.

The convergence rates of the different scans varied from one scan of the boundaries to another in Pearson's study. This study showed a convergence rate faster for the spiral scan rather than the modified serial scan. The determination that certain types, magnitudes, and ratios of boundary conditions can cause one boundary scan to converge faster than another could form an area of further study of the scanning technique.

### Bibliography

1. Pearson, J. T. "Comparison of Point and Block Successive Over-relaxation in the Solution of Laplace's Equation in Two Dimensions." Masters thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1969.
2. Cudahy, G. F. "Investigation of Accelerating the Convergence of an Implicit Numerical Solution of Transient Heat Transfer Problems." Masters thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, August 1965.
3. Wright, K. D. "Investigation of Block Overrelaxation for Accelerating the Convergence of the Implicit Numerical Solution of the Transient Heat Transfer Equation." Masters thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1967.
4. Nakamura, S. Computation Methods in Engineering and Science. New York: John Wiley & Sons, 1977.
5. Huebner, K. H. The Finite Element Method for Engineers. New York: John Wiley & Sons, 1975.
6. Myers, G. E. Analytical Methods in Conduction Heat Transfer. New York: McGraw-Hill, Inc., 1971.
7. Clark, M. and Hansen, K. F. Numerical Methods of Reactor Analysis. New York: Academic Press, 1964.
8. Varga, R. S. Matrix Iterative Analysis. New Jersey: Prentice-Hall, Inc., 1962.
9. Wachspress, E. L. Iterative Solution of Elliptic Systems. New Jersey: Prentice-Hall, Inc., 1966.
10. Ketter, R. L. and Praebel, S. P. Modern Methods of Engineering Computation. New York: McGraw-Hill, Inc., 1969.
11. Faddeev, D. K. and Faddeeva, V. N. Computational Methods in Linear Algebra. San Francisco: W. H. Freeman and Company, 1963.

## Appendix A

### Sample Coarse Mesh Rebalancing

Given the region shown in Figure 5, a coarse mesh is imposed and the points partitioned by four partitioning matrices.

$$P_1 = \text{diag} \{1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0\} \quad (85)$$

$$P_2 = \text{diag} \{0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0\} \quad (86)$$

$$P_3 = \text{diag} \{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0\} \quad (87)$$

$$P_4 = \text{diag} \{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1\} \quad (88)$$

These satisfy the requirement that the partitioning matrices sum to the identity matrix.

The weighting vectors are chosen as

$$w_i = P_i \underline{1} \quad (82)$$

The weighting vectors have the same elements as the main diagonals of the corresponding partitioning matrices in this case.

A matrix for the fine mesh problem could be

$$\begin{bmatrix} 4 & -1 & & & & & & & & & & & & & \\ -1 & 4 & -1 & & & & & & & & & & & & \\ & -1 & 4 & -1 & & & & & & & & & & & \\ & & -1 & 4 & & & & & & & & & & & \\ -1 & & & & 4 & -1 & & & & & & & & & \\ & -1 & & & -1 & 4 & -1 & & & & & & & & \\ & & -1 & & -1 & -1 & 4 & -1 & & & & & & & \\ & & & -1 & & -1 & -1 & 4 & & & & & & & \\ & & & & -1 & & -1 & -1 & 4 & -1 & & & & & \\ & & & & & -1 & & -1 & -1 & 4 & -1 & & & & \\ & & & & & & -1 & & -1 & -1 & -1 & 4 & & & \\ & & & & & & & -1 & & -1 & -1 & -1 & 4 & & \\ & & & & & & & & -1 & & -1 & -1 & -1 & 4 & \\ & & & & & & & & & -1 & & -1 & -1 & -1 & 4 \end{bmatrix} = A \quad (90)$$

The given matrix  $\underline{A}$  is multiplied by each partitioning matrix.

$$\begin{bmatrix} 4 & -1 & & & & & & & & & \\ -1 & 4 & & & & & & & & & \\ & -1 & 0 & & & & & & & & \\ & & & 0 & & & & & & & \\ -1 & & & & 4 & -1 & & & & & \\ & -1 & & & -1 & 4 & & & & & \\ & & & & & & 0 & & & & \\ & & & & & & & 0 & & & \\ & & & & -1 & & & & 0 & & \\ & & & & & -1 & & & & 0 & \\ & & & & & & & 0 & & & 0 \\ & & & & & & & & 0 & & \\ & & & & & & & & & 0 & \\ & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \end{bmatrix} = \underline{A} \underline{P} \quad (91)$$

The four different matrix products are each multiplied by the current iteration vector. For example,

$$\underline{A} \underline{P} \underline{t} = \begin{bmatrix} 4 T_1 - T_2 - T_5 \\ -T_1 + 4 T_2 - T_6 \\ -T_3 \\ 0 \\ -T_1 + 4 T_5 - T_6 - T_9 \\ -T_2 - T_5 + 4 T_6 - T_{10} \\ -T_6 \\ 0 \\ -T_5 \\ -T_6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (92)$$

The required scalar multiplication is then carried out with each of the four vectors and each of the four weighting vectors.

$$\begin{aligned} \langle w_i, A P_i t_i \rangle &= (4 t_i - t_2 - t_5) + (- t_i + 4 t_2 - t_6) + \\ &(- t_i + 4 t_5 - t_6 - t_9) + (- t_2 - t_5 + 4 t_6 - t_{10}) \end{aligned} \quad (93)$$

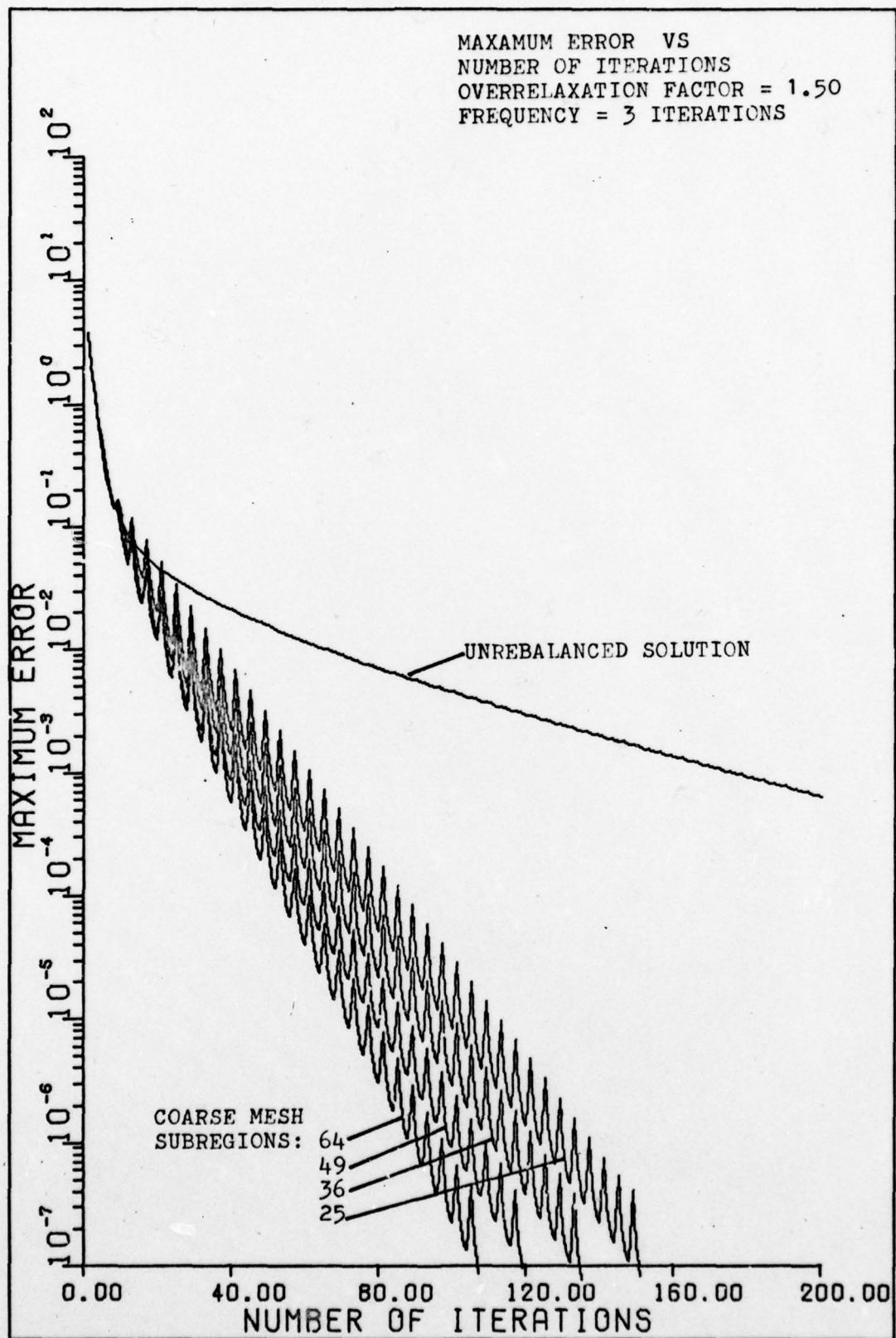
This number forms one of the 16 elements of an  $4 \times 4$  matrix which will be used to determine the rebalancing factors. Each product of  $w_k$  and  $A P_i T$  yields the element  $B_{k,i}$  of the new matrix which will be called  $B$ .

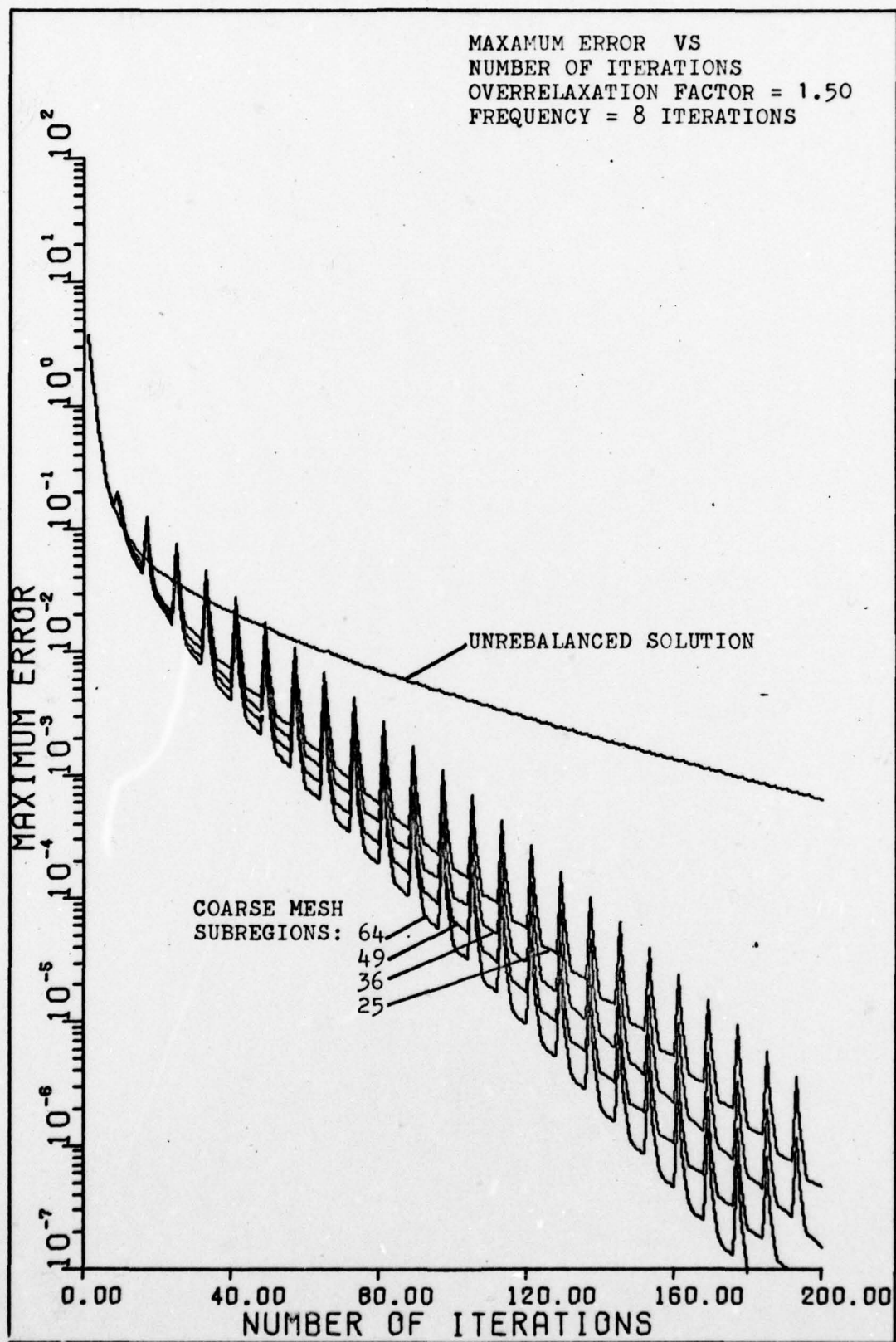
Scalar multiplication is then carried out with the weighting vectors and the known vector. These multiplications yield four elements for a new vector,  $G$ , which will be the known vector in the coarse mesh equation.

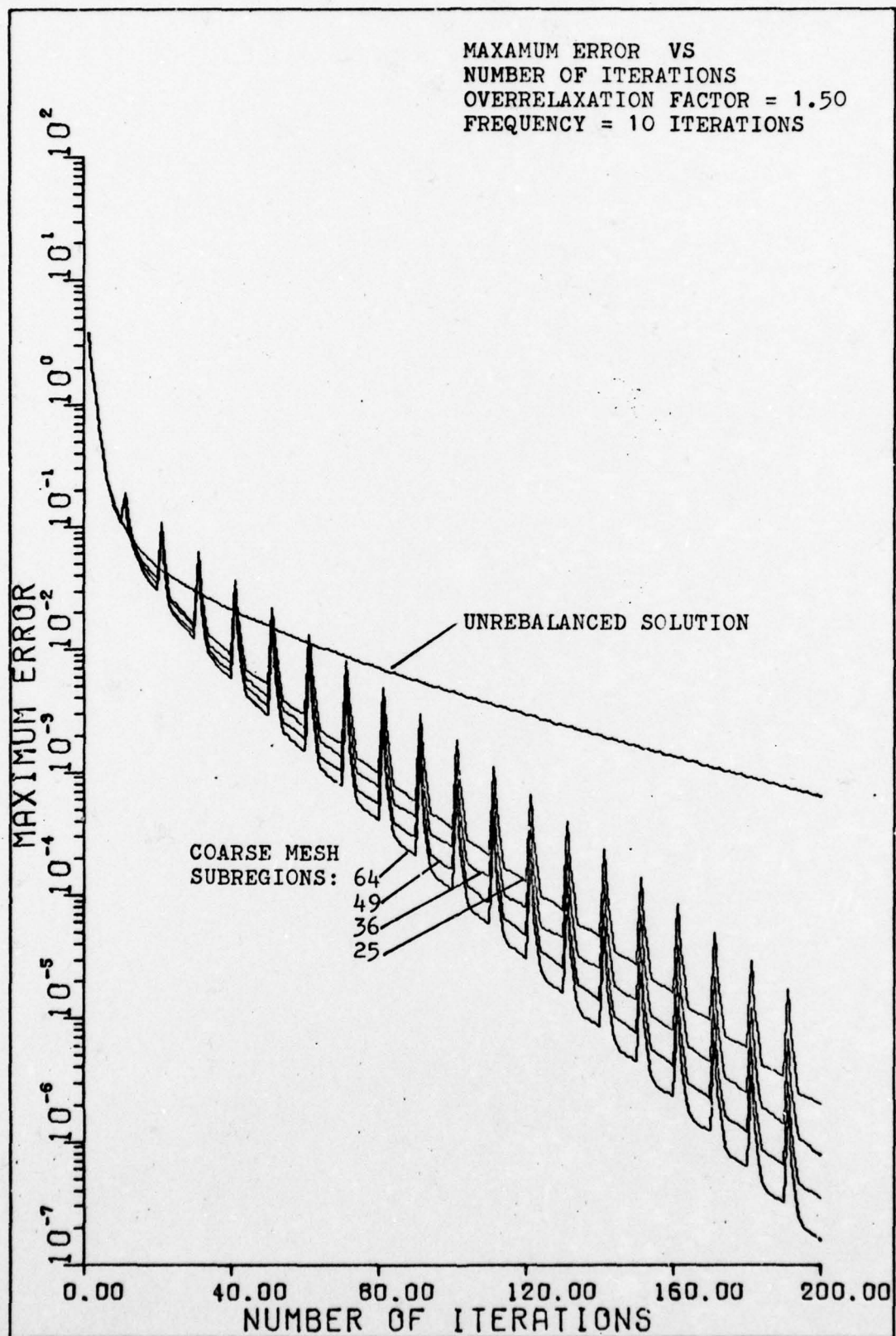
A matrix equation has been formed. A  $4 \times 4$  matrix and a vector of length 4 have been formed. The solution of this matrix equation provides another vector of length 4. This vector contains the rebalancing coefficients, each of which corresponds to one subregion in the coarse mesh. Each component of the iteration vector is multiplied by the corresponding rebalancing factor and rebalancing is done. The new iteration vector is returned to the iteration scheme and iteration continued until convergence occurs or another rebalancing is done.

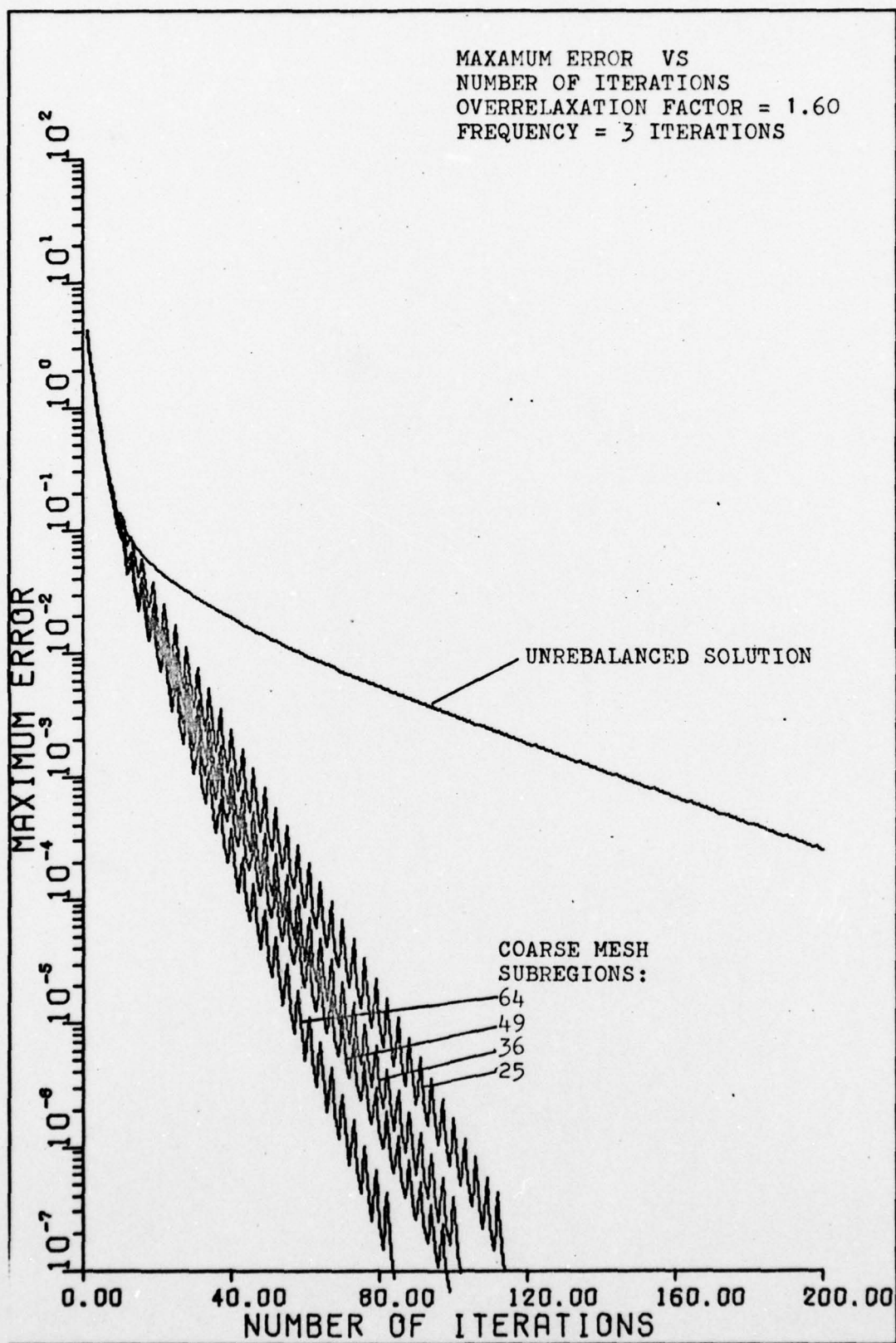
Appendix B

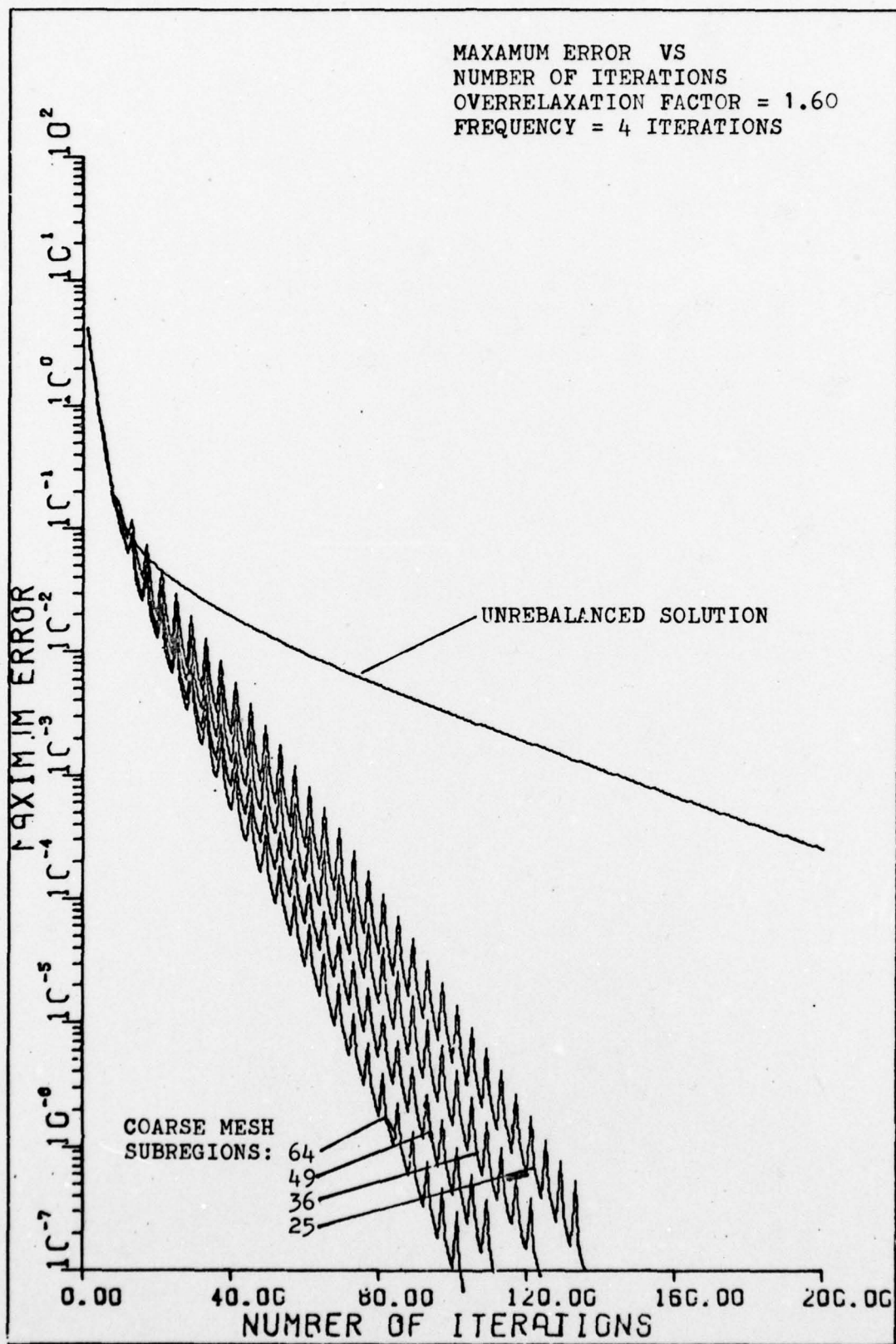
Plots of Maximum Error vs Number of Iterations

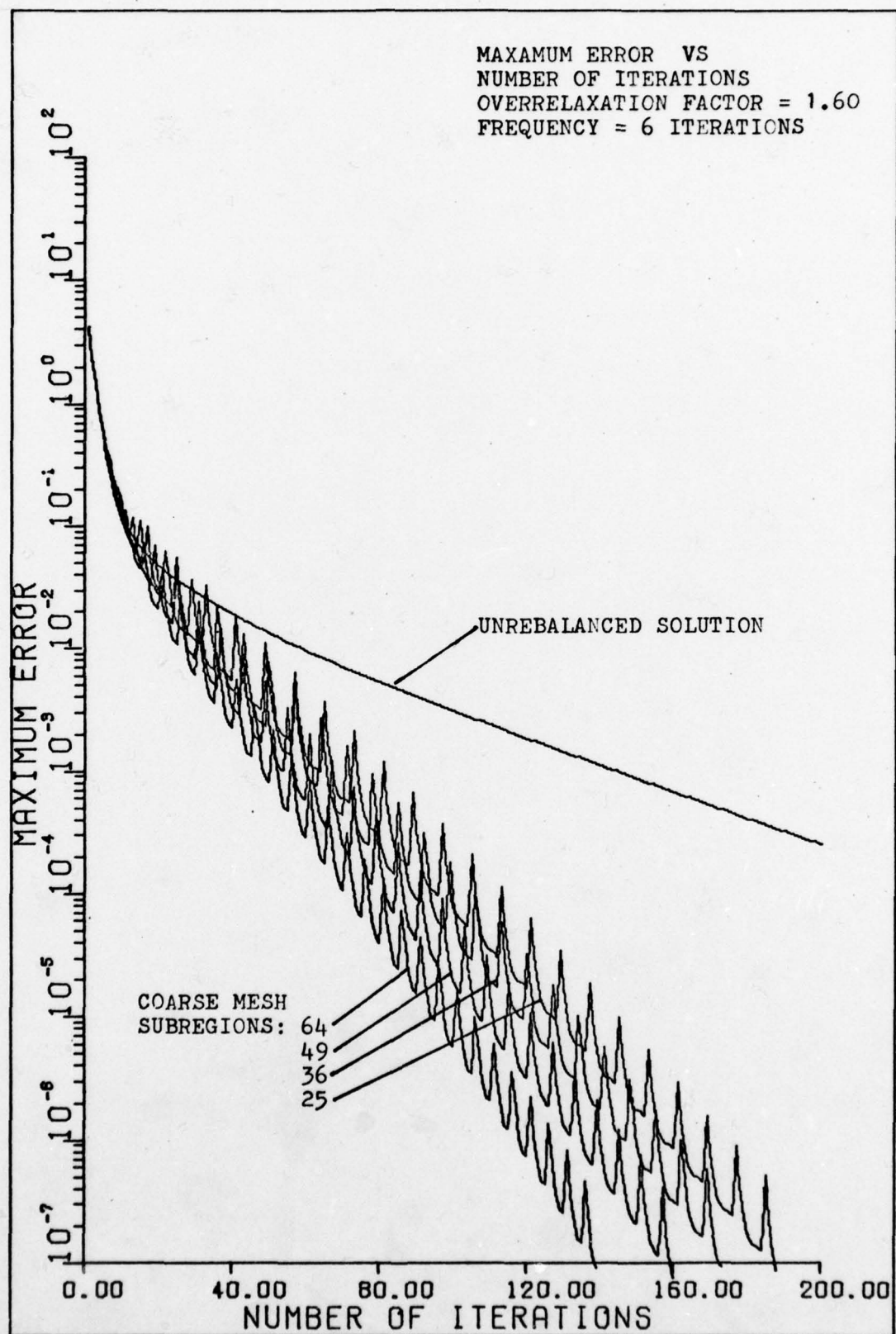


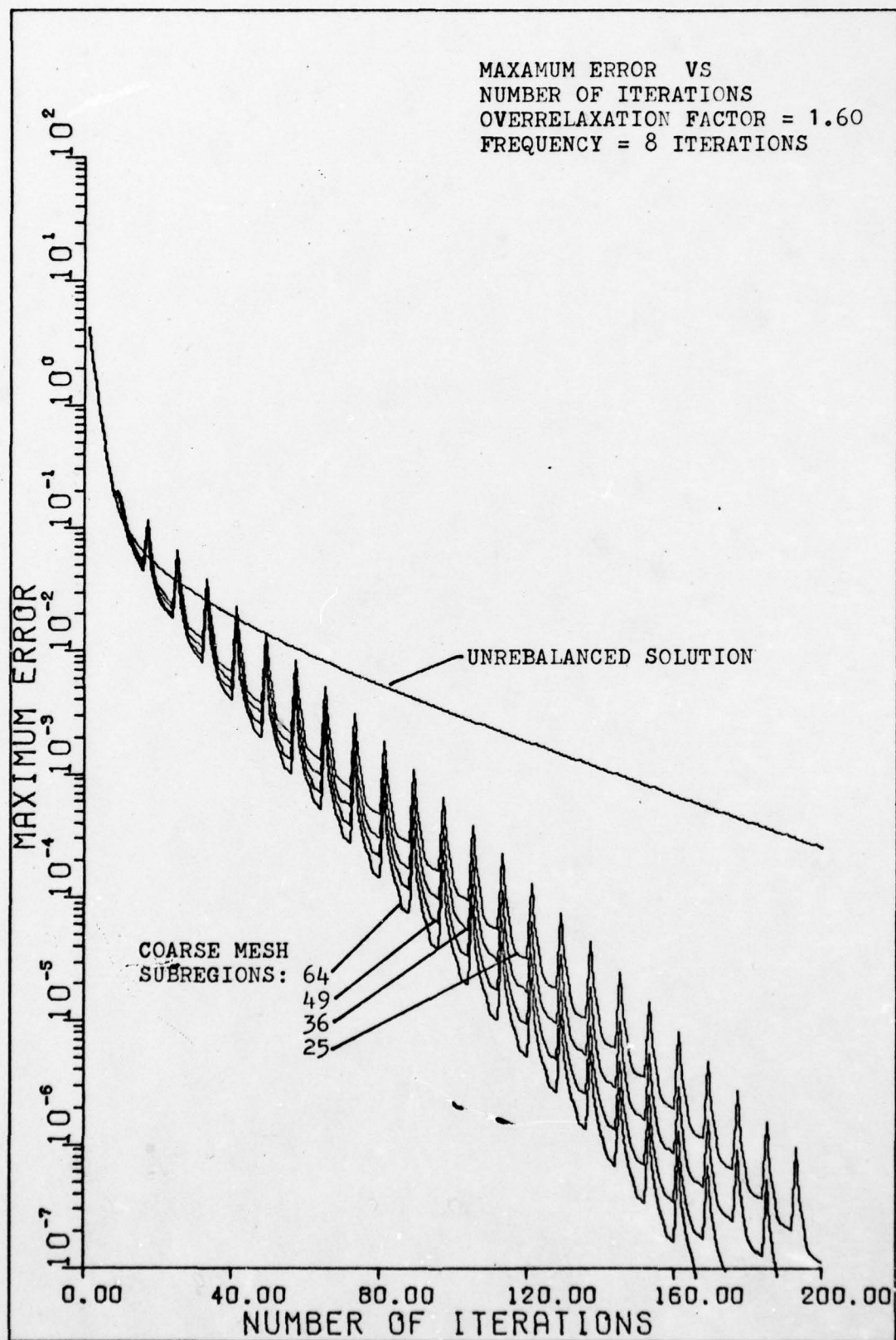


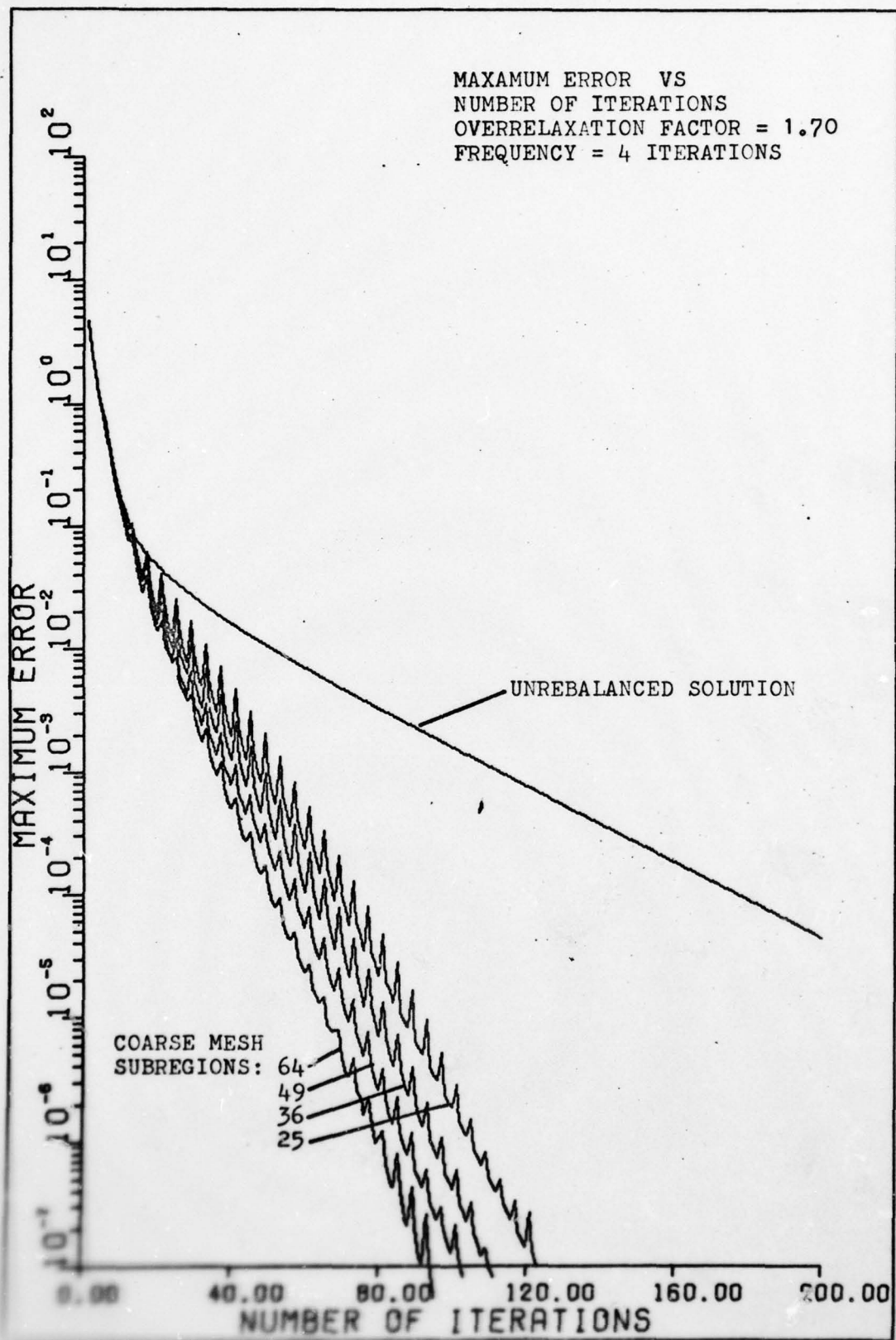


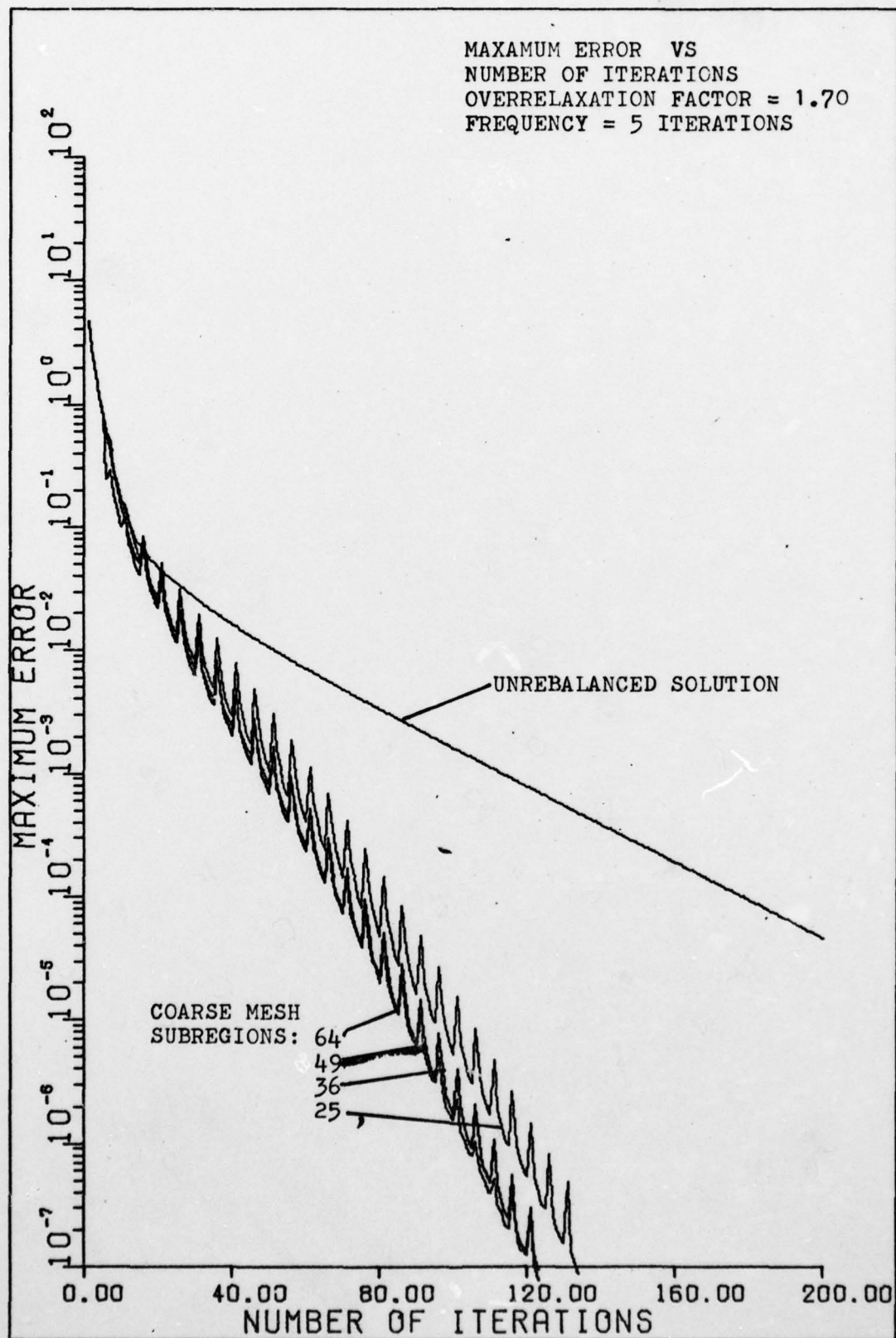


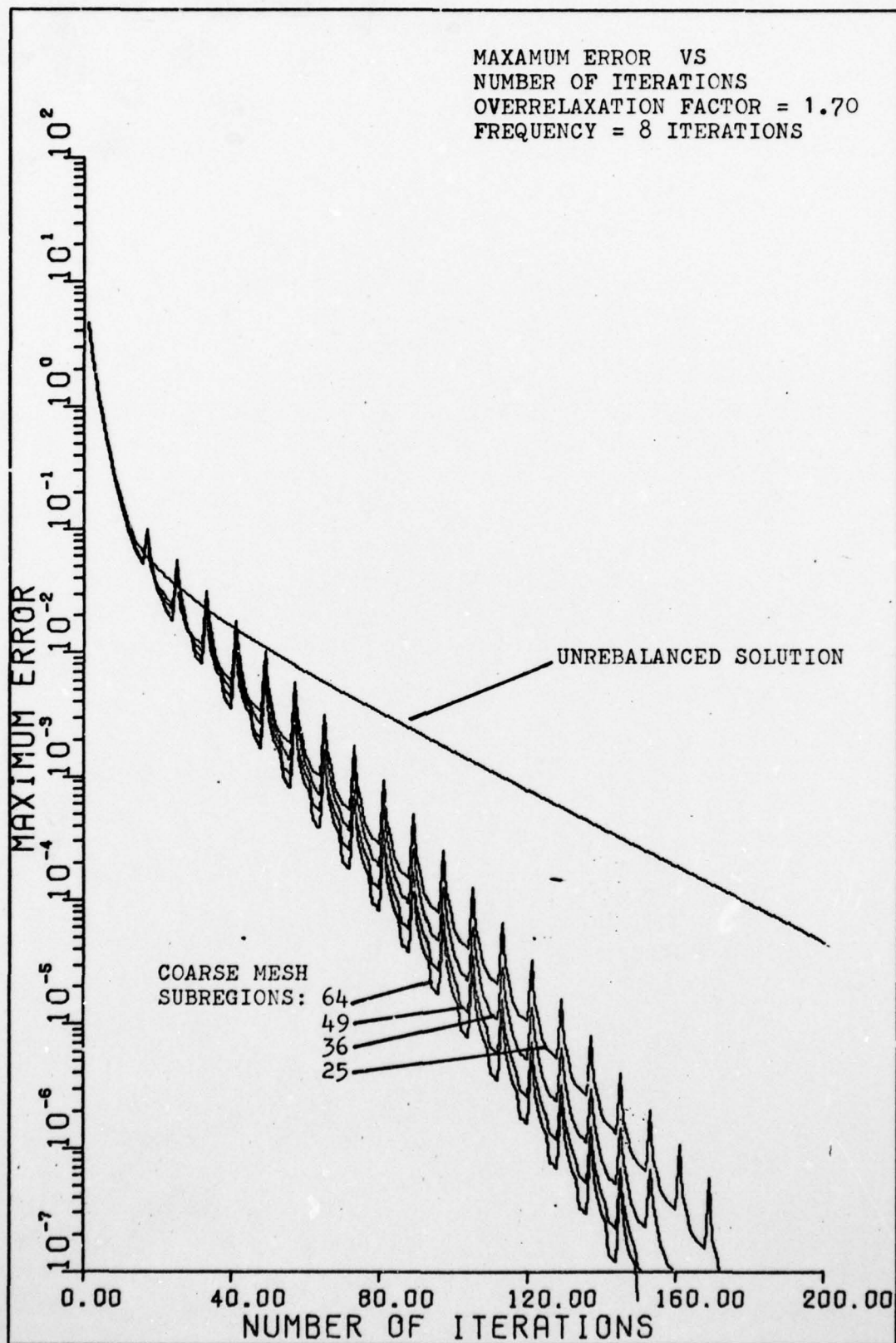


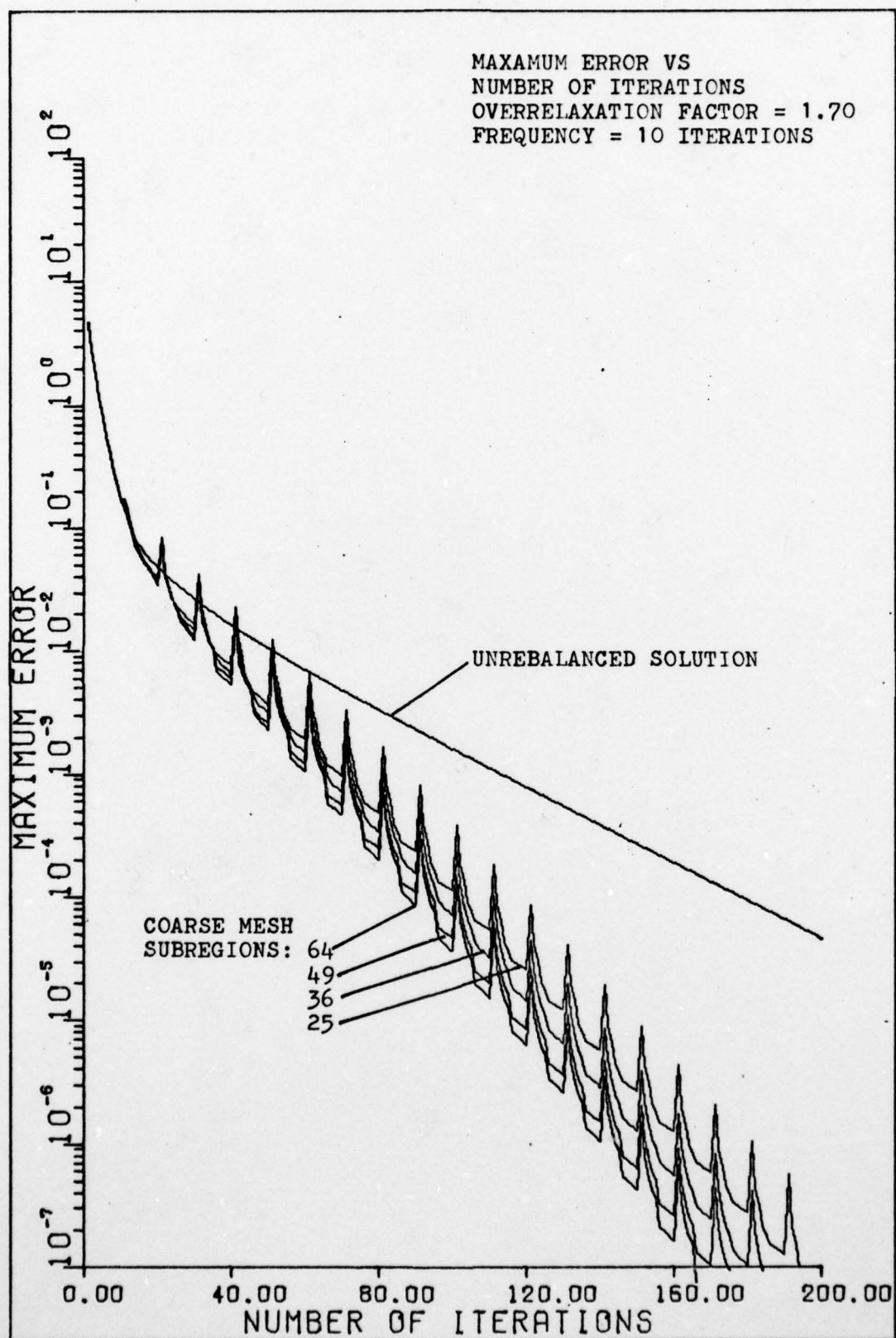


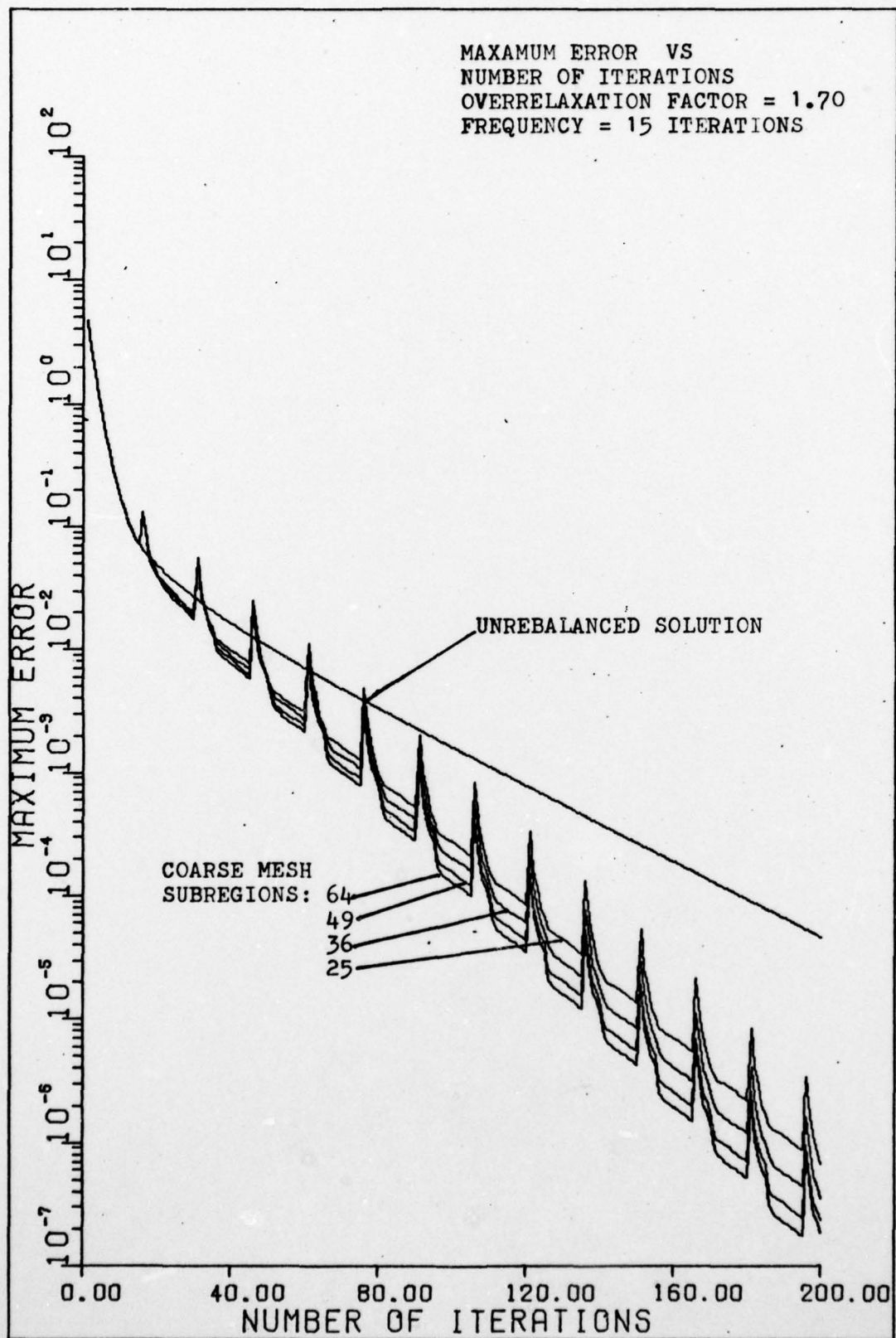


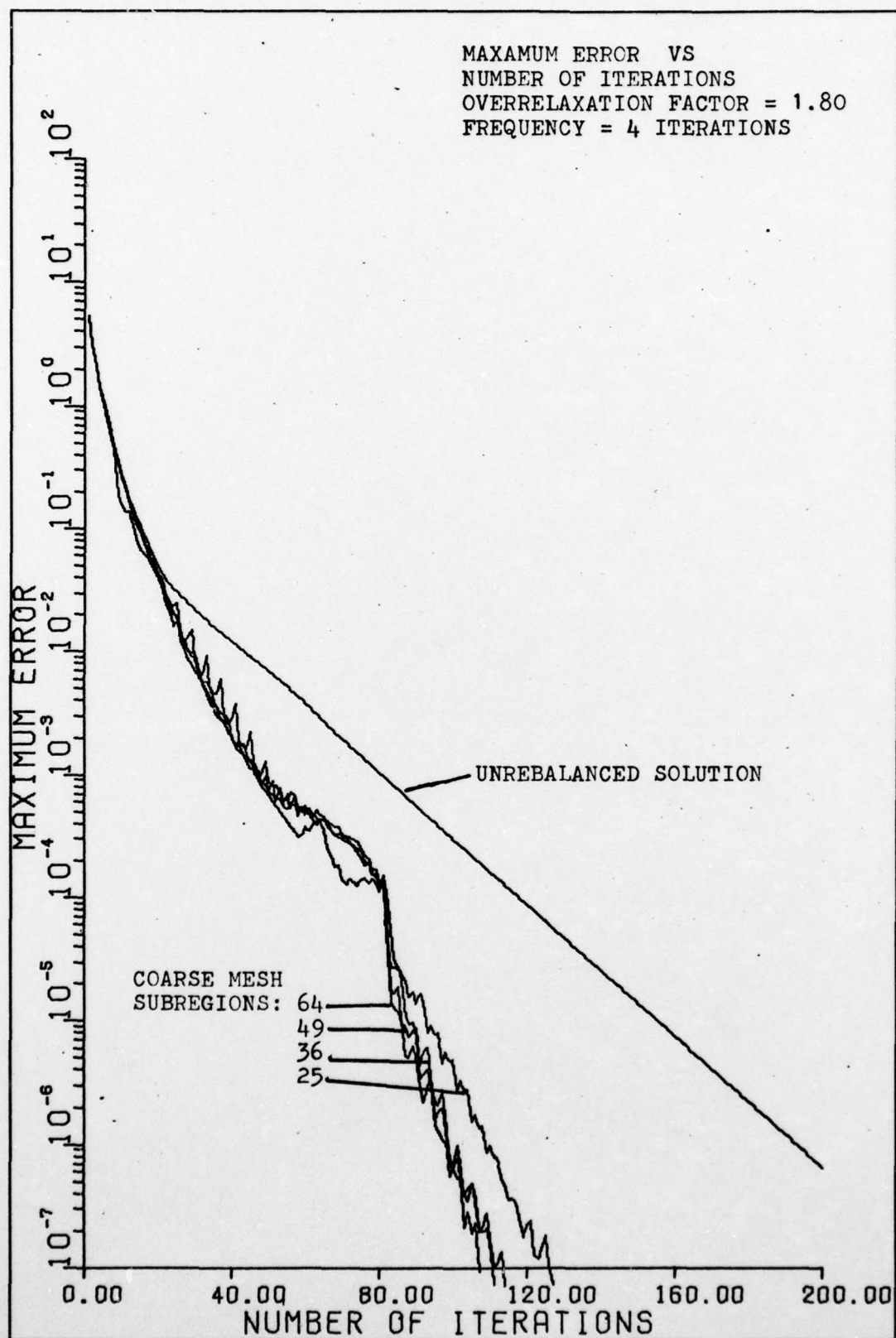


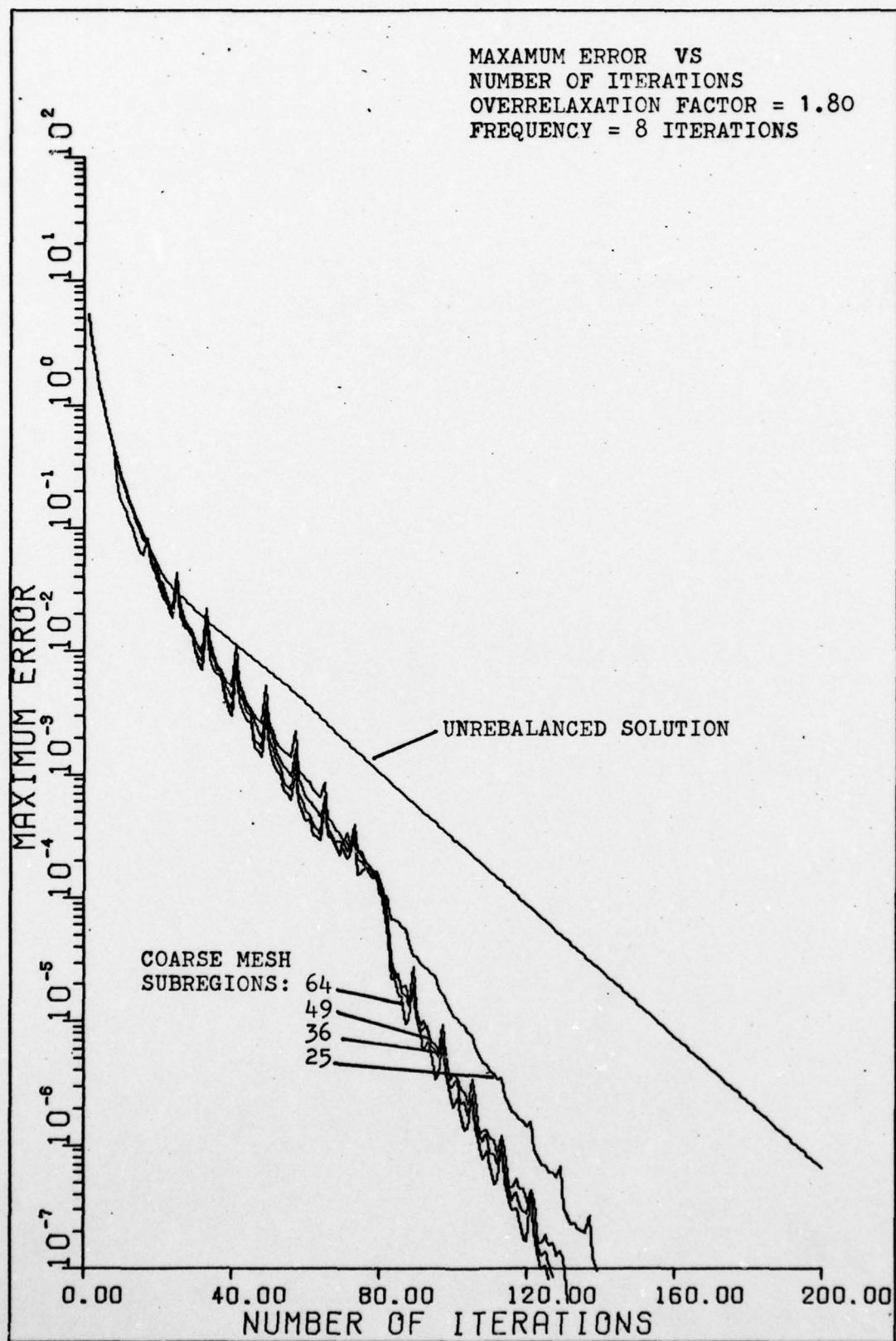


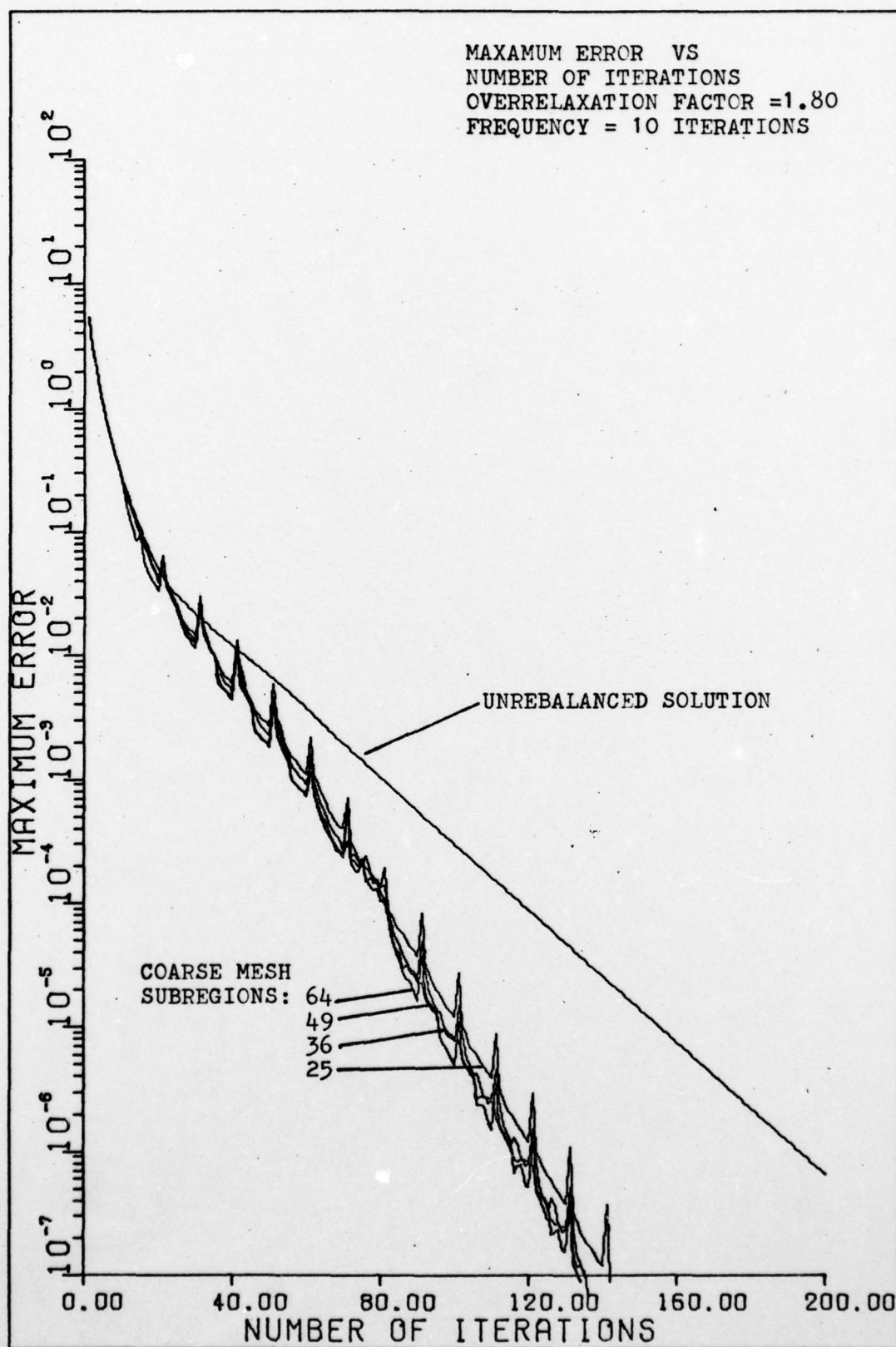


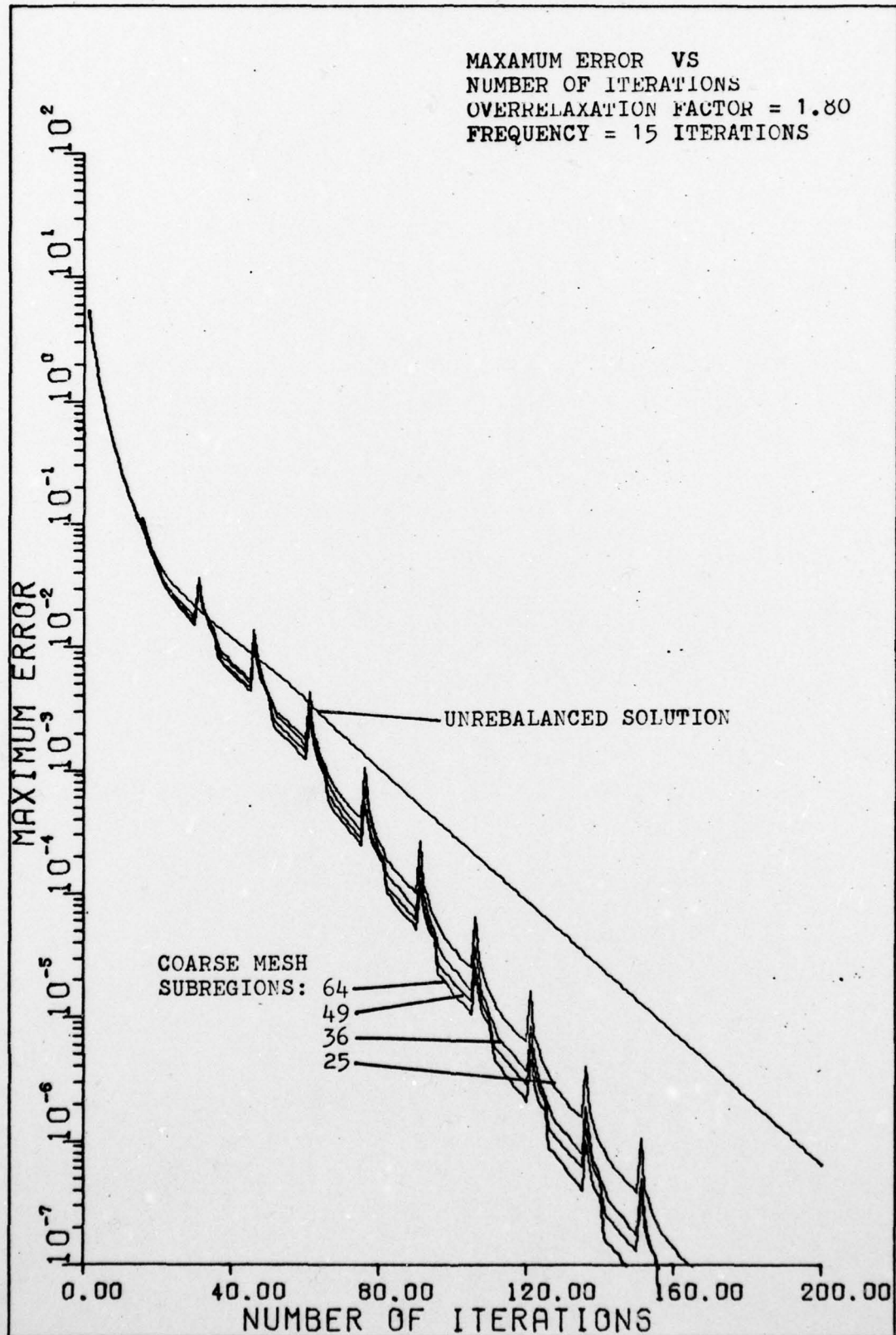


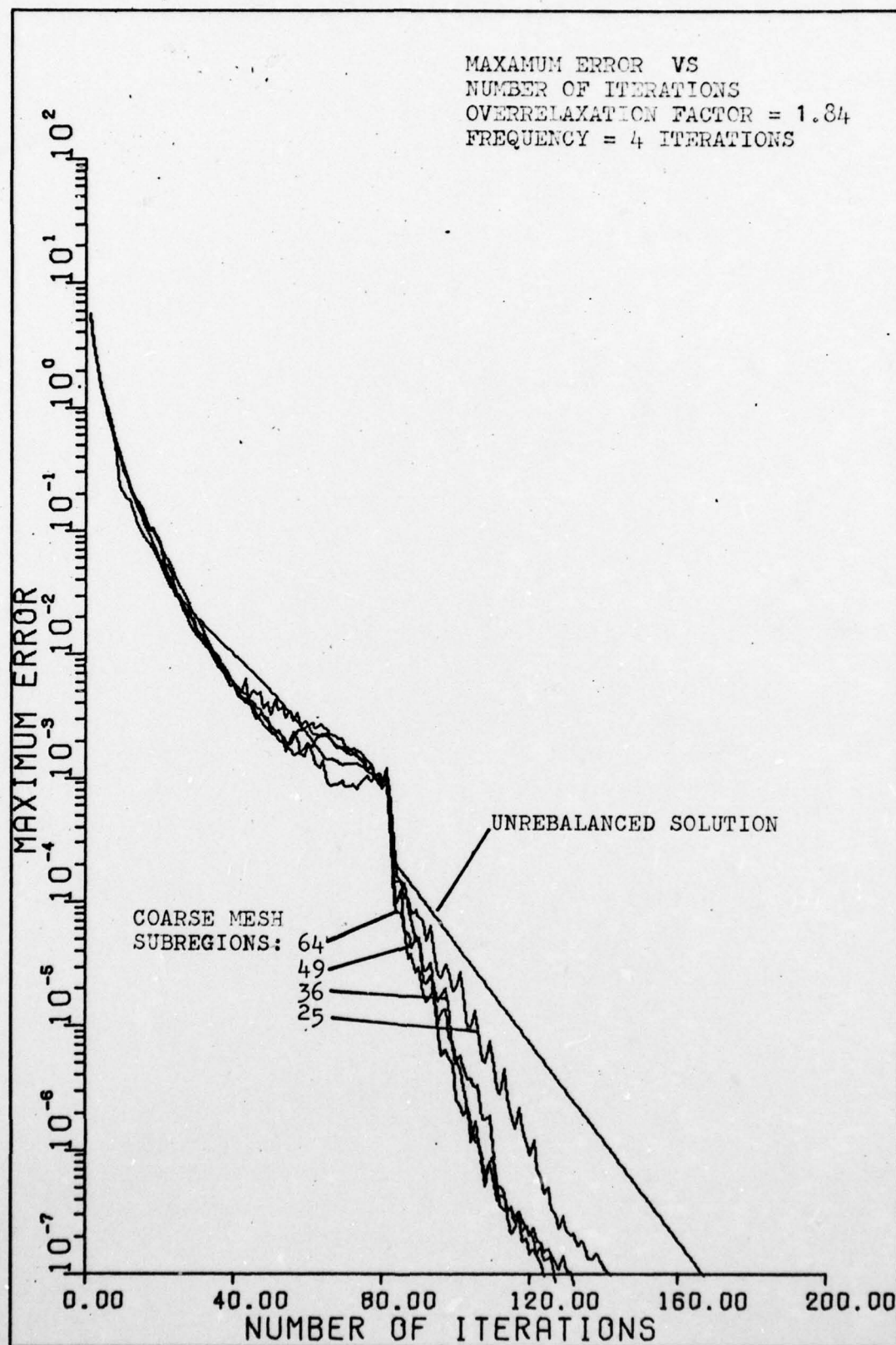


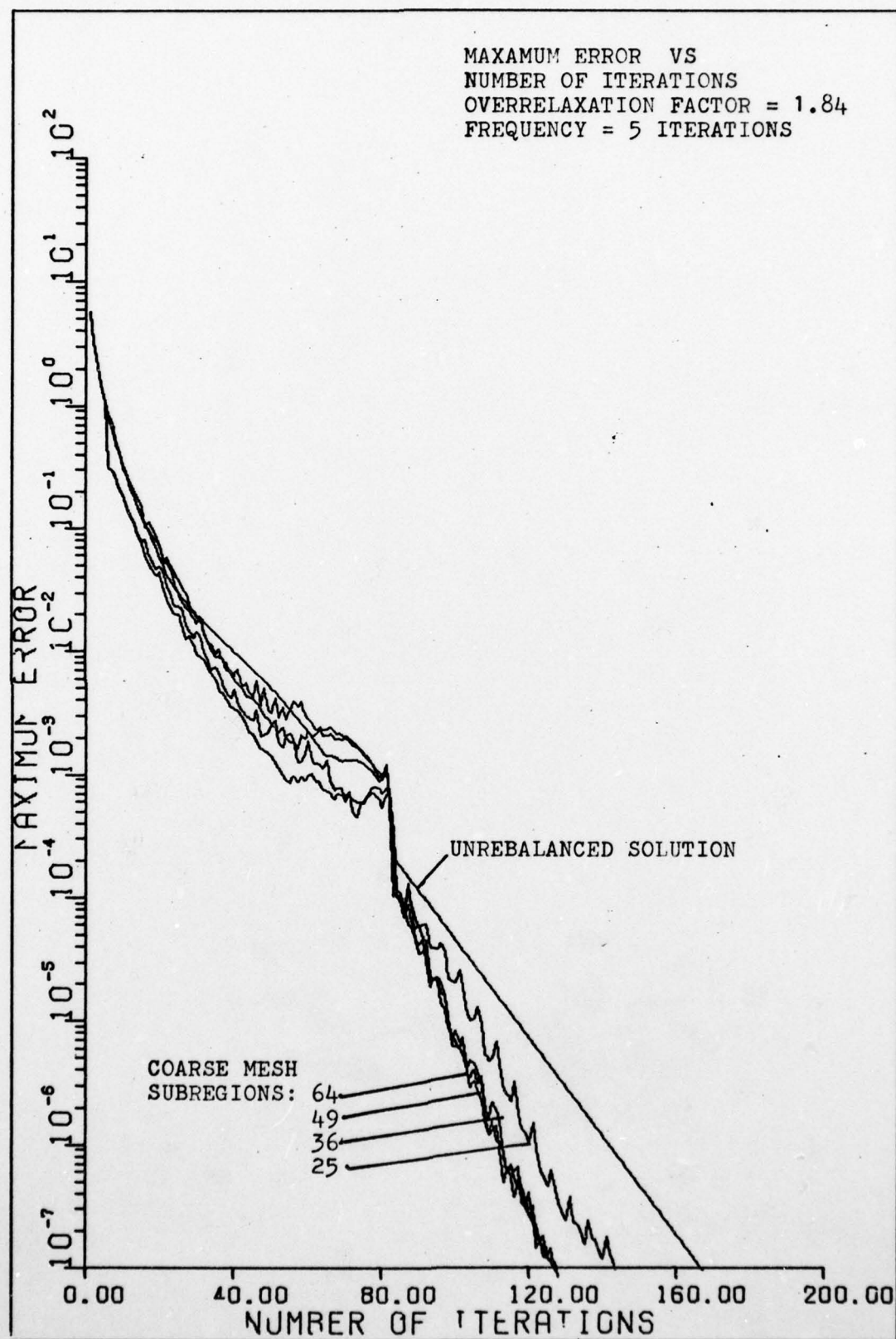


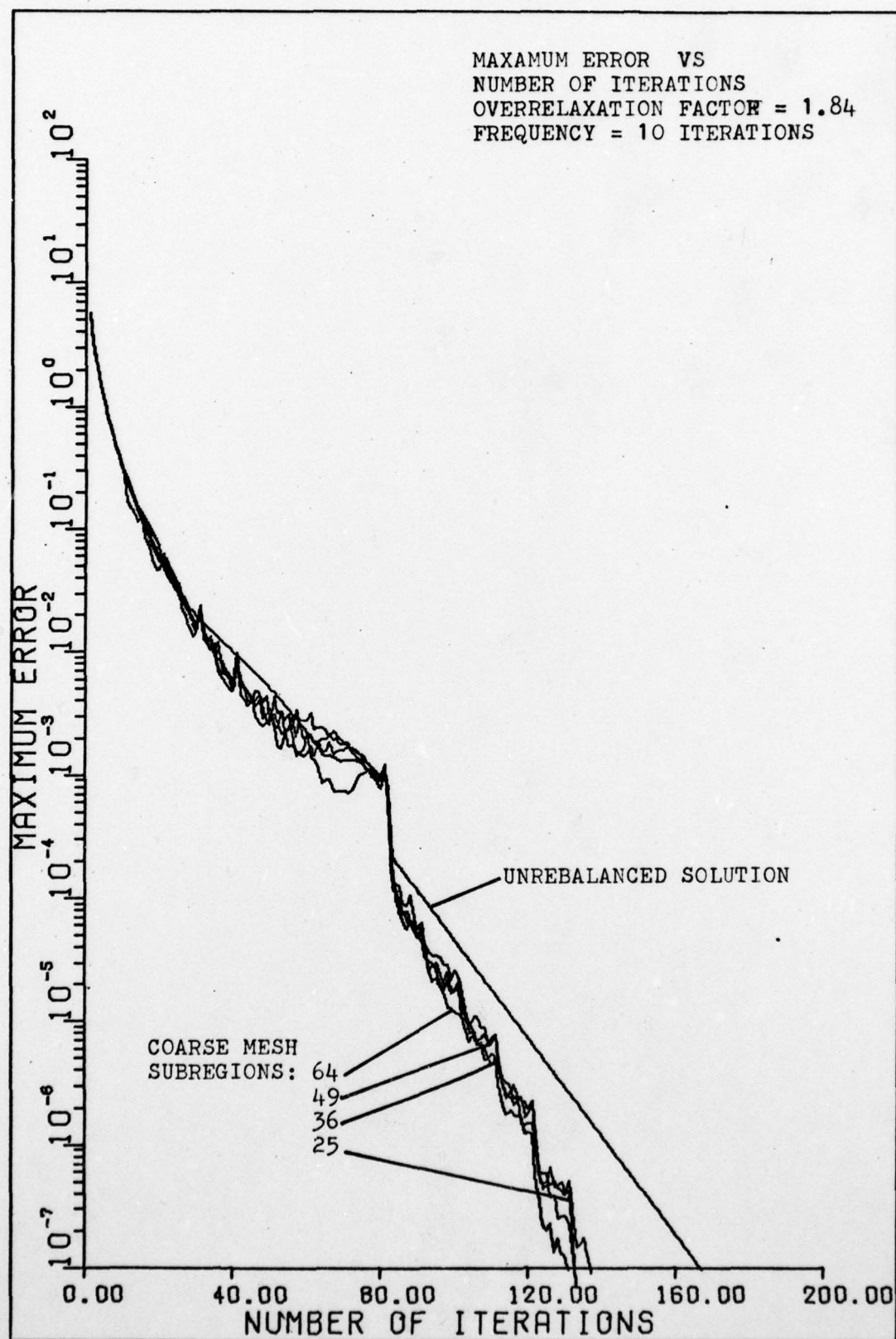












Vita

Frederick Jacob Jaeger was born on 12 September 1949 in Jersey City, New Jersey. He graduated from Saint Peter's Preparatory School, Jersey City, New Jersey in June 1967. In September 1967, he entered the Newark College of Engineering, Newark, New Jersey. Upon graduation from the Newark College of Engineering in June 1971 with a degree of Bachelor of Science in Mechanical Engineering, Captian Jaeger was commissioned a Second Lieutenant in the United States Air Force. He was assigned to the Air Force Flight Test Center, Edwards Air Force Base, California as a flight test engineer from July 1971 until August 1975. From September 1975 until May 1977, he served as a design engineer at the Air Force Cryptological Depot, San Antonio, Texas. In June 1977, he entered the graduate program in Nuclear Engineering at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

Permanent address: 30 Stag Street  
Jersey City, New Jersey

=Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GNE/PH/78D-6	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INVESTIGATION OF ACCELERATING THE FINITE ELEMENT SOLUTION OF THE TWO DIMENSIONAL STEADY STATE HEAT TRANSFER EQUATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) Frederick J. Jaeger Captain, USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Materials Laboratory (AML/MBC) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1978
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 JOSEPH P. HIPPS, MAJOR, USAF Director of Information 1-23-79		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) successive overrelaxation coarse grid rebalancing acceleration parameter overrelaxation factor coarse mesh rebalancing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The relative solution time is studied for two methods of accelerating successive overrelaxation. Reordering of equations by nodal point scanning, and coarse mesh rebalancing are used. The finite element solution of the steady-state two-dimensional heat transfer equation is used to test these methods. Scanning the boundary nodal points first was found to reduce the number of iterations necessary for convergence by up to 13%, but computer execution		

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A064 417

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 20/13  
INVESTIGATION OF ACCELERATING THE FINITE ELEMENT SOLUTION OF TH--ETC(U)  
DEC 78 F J JAEGER

UNCLASSIFIED

AFIT/ONE/PH/78D-6

NL

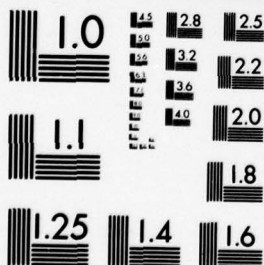
2 OF 2  
AD  
A064 417



END  
DATE  
FILMED  
4 -79  
DDC

Unclass  
SECURITY CLASSIFICATION

20. was increase  
the solution  
the number o  
execution ti  
to 30% of th  
was found to  
fast as the  
with rebalan  
proper coars



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. was increased by up to 7%. Coarse mesh rebalancing was found to speed the solution with arbitrary successive overrelaxation factor by reducing the number of iterations to 15% of that without rebalancing. The computer execution time required for a well chosen coarse mesh was only reduced to 30% of that without rebalancing. The successive overrelaxation factor was found to influence the optimum rebalancing frequency. Solutions as fast as the solution with the optimum overrelaxation factor were obtained with rebalancing and arbitrary overrelaxation factors. Rules for a proper coarse mesh and rebalancing frequency are given.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)